Umělá inteligence v kontextu kybernetické bezpečnosti subjektu

Bc. Zbyněk Šalomon

Diplomová práce
2023

Univerzita Tomáše Bati ve Zlíně
Fakulta logistiky a krizového řízení

Univerzita Tomáše Bati ve Zlíně
Fakulta logistiky a krizového řízení
Ústav ochrany obyvatelstva

Akademický rok: 2022/2023

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Bc. Zbyněk Šalomon**
Osobní číslo:	**L21324**
Studijní program:	**N1032A020002 Bezpečnost společnosti**
Specializace:	**Ochrana obyvatelstva**
Forma studia:	**Kombinovaná**
Téma práce:	**Umělá inteligence v kontextu kybernetické bezpečnosti subjektu**

Zásady pro vypracování

1. Proveďte rešerši současného stavu umělé inteligence v kontextu kybernetické bezpečnosti subjektu.
2. Analyzujte problematiku využití umělé inteligence v kontextu kybernetické bezpečnosti subjektu.
3. Analyzujte možnosti využití vybraných algoritmů umělé inteligence ve smyslu zvýšení úrovně zabezpečení subjektu.
4. Navrhněte implementaci vybraných prvků umělé inteligence do systému zabezpečení subjektu.

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. GOODFELLOW, Ian, Yoshua BENGIO a Aaron COURVILLE. *Deep Learning* [online]. Cambridge: MIT Press, 2016. ISBN 0262035618. Dostupné z: https://www.deeplearningbook.org/.
2. KOLOUCH, Jan a Pavel BAŠTA. *CyberSecurity* [online]. Praha: CZ.NIC, z.s.p.o. CZ.NIC, 2019. ISBN 978-80-88168-31-7.
3. PEDRYCZ, Witold a Shyi-Ming CHEN, ed. *Deep learning: algorithms and applications*. Cham: Springer. Studies in computational intelligence, 2020. ISBN 978-3-030-31759-1.

Další odborná literatura dle doporučení vedoucího diplomové práce.

Vedoucí diplomové práce: **Ing. Petr Svoboda, Ph.D.**
Ústav ochrany obyvatelstva

Datum zadání diplomové práce: **1. prosince 2022**
Termín odevzdání diplomové práce: **28. dubna 2023**

L.S.

doc. Ing. Zuzana Tučková, Ph.D.
děkanka

prof. Ing. Dušan Vičar, CSc.
ředitel ústavu

V Uherském Hradišti dne 2. prosince 2022

PROHLÁŠENÍ AUTORA
DIPLOMOVÉ PRÁCE

Beru na vědomí, že:

- diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému a dostupná k nahlédnutí;
- na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- podle § 60 odst. 1 autorského zákona má Univerzita Tomáše Bati ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tj. k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou obsahově totožné.

V Uherském Hradišti, dne: 28.4.2023

Jméno a příjmení studenta: Bc. Zbyněk Šalomon

.....
podpis studenta

ABSTRAKT

Tato diplomová práce se zaměřuje na využití umělé inteligence (AI) v kontextu kybernetické bezpečnosti subjektů s důrazem na zabezpečení kamerových dohledových systémů. V teoretické části práce jsou podrobně popsány základní pojmy AI, vývoj AI a současný stav využití AI v kybernetické bezpečnosti. Dále se teoretická část práce věnuje kamerám a zabezpečení kamerových systémů.

Praktická část práce se zaměřuje na návrh algoritmů v jazyce Python, které mají zvýšit zabezpečení kamerového dohledového systému. Tyto algoritmy jsou navrženy tak, aby dokázaly rozpoznat a identifikovat potenciální hrozby a vyhodnotit jejich váhu a riziko. Následně byly podrobeny testování funkčnosti v běžném provozu.

Výsledky testování algoritmů ukazují, že jejich použití může výrazně zlepšit ochranu kamerového systému a snížit riziko kybernetických útoků. Nicméně, přínos využití AI v kontextu kybernetické bezpečnosti kamerových systémů ještě není plně jistý a může být ovlivněn mnoha faktory.

Klíčová slova: algoritmy, video dohledové systémy, kyberbezpečnost, umělá inteligence

ABSTRACT

This thesis focuses on the use of artificial intelligence (AI) in the context of cyber security of entities, with an emphasis on securing a camera surveillance system. The theoretical part of the thesis describes in detail the basic concepts of AI, the development of AI and the current state of the use of AI in cyber security. The theoretical part of the thesis also focuses on cameras and the security of camera systems.

The practical part of the thesis focuses on designing algorithms in Python that aim to increase the security of a camera surveillance system. These algorithms are designed to recognize and identify potential threats and evaluate their weight and risk. One of the goals of the thesis was to test the functionality of these algorithms in regular operation.

The results of the algorithm testing demonstrate that their use can significantly improve the security of the camera system and reduce the risk of cyberattacks. However,

the benefits of using AI in the context of cyber security of camera systems are not yet fully certain and can be influenced by many factors.

Keywords: Algorithms, Artificial Intelligence, Cybersecurity, Video Surveillance System

Mé poděkování patří Ing. Petru Svobodovi Ph.D. za odborné vedení, které mi v průběhu zpracování diplomové práce věnoval. Zároveň bych chtěl poděkovat mé rodině a zejména manželce Gabriele za podporu, kterou mi poskytla při studiu.

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

CÍL A METODY ZPRACOVÁNÍ

Tato diplomové práce se bude zabývat návrhy algoritmů umělé inteligence vedoucí ke zvýšení úrovně zabezpečení video dohledového systému subjektu. Tyto systémy jsou stejně jako ostatní prvky kategorie „*Internetu věcí*“ častým místem průniku do sítě subjektu, a proto je žádoucí se na jejich bezpečnost zaměřit.

Diplomová práce si klade za cíl přinést nový pohled na využití umělé inteligence v kybernetické bezpečnosti kamerových systémů, na její příspěvek ke komplexnímu kybernetickému zabezpečení subjektu a ilustrovat její potenciál pro budoucí výzkum v této oblasti. Výsledky mohou sloužit jako základ pro další výzkum a implementaci algoritmů AI pro zvýšení bezpečnosti kamerových systémů.

Hlavním cílem diplomové práce tedy je:

Návrh algoritmů umělé inteligence implementovatelných do vybraného prvku bezpečnostního systému kybernetické bezpečnosti.

Dílčími cíli práce pak jsou:

- Rešerše problematiky umělé inteligence v kontextu kybernetické bezpečnosti subjektu.
- Návrh algoritmů umělé inteligence pro implementaci do vybraného prvku.
- Ověření funkcionality navržených algoritmů.

Cílů je dosaženo pomocí využití metod:

- Analýza dat: Tato metoda se používá k porozumění datům, která jsou získána z výstupu navržených algoritmů. Zahrnuje sběr dat, analýzu a interpretaci výsledků, aby se identifikovaly oblasti, které potřebují zlepšení.
- Komparace: sloužila ke zjišťování rozdílů mezi jednotlivými programy a procesy popisovaných v průběhu diplomové práce.
- Dedukce byla zvolena pro popis konkrétních prvků systému, k identifikaci specifických požadavků a omezení pro konkrétní aplikaci, což by mohlo vést k výběru nejvhodnějších algoritmů pro řešení daného problému.

ÚVOD

Kybernetická bezpečnost je oblast informatiky, která se vlivem posledních let dostala do popředí zájmu. Mohou za to bohužel především jedinci, případně celé skupiny, které se specializují na kybernetickou kriminalitu. Tato oblast kriminality je, vzhledem k rozšíření využívání výpočetních technologií, mnohdy mnohem nebezpečnější než kriminalita fyzická. Kybernetická kriminalita – na rozdíl od té fyzické – totiž nezná hranic a běžně se stává, že si příležitostný uživatel ani nevšimne, že on sám je nebo byl obětí nějakého kyberútoku. A nemusí jít nutně jen o snahu získat data či prostředky oběti. Rozšířený je zejména fenomén tzv. „zombie počítačů“, kdy je počítač napaden a bez vědomí majitele používán někým zvenčí k rozesílání spamu, DDoS útokům apod.

Jako v mnoha dalších oblastech, i zde umělá inteligence otevírá množství příležitostí. Bohužel oběma stranám barikády. Může pomoci s návrhem bezpečnostních struktur, ale také s jejich překonáváním. Vše je jen otázkou dat zadaných do jejich algoritmů.

Tato práce se bude zabývat možnostmi využití umělé inteligence ke zvýšení nejen kybernetického, ale i fyzického zabezpečení subjektu.

I. TEORETICKÁ ČÁST

1 POJMOVÝ APARÁT

Pod samotným pojmem umělé inteligence se skrývá řada pojmů a jejich neznalost komplikuje orientaci v celé problematice. Následující terminologický výčet byl vybrán vzhledem k tomu, že se s ním bude v práci operovat a přispěje k orientaci dále v textu.

- ***Algoritmus***: Postup nebo krok za krokem pro řešení úkolu nebo problému.
- ***Algoritmy umělé inteligence***: Jedná se o matematické modely a postupy použité k řešení úloh v oboru umělé inteligence.
- ***Regrese***: Typ strojového učení, který se používá k vytváření předpovědního modelu na základě historických dat. Cílem je předpovědět hodnotu určité veličiny v budoucnosti na základě vztahu mezi touto veličinou a jinými relevantními faktory v minulosti. Například, při vytváření předpovědního modelu ceny akcií, můžeme použít regresní analýzu k vytvoření modelu, který na základě historických dat předpovídá budoucí cenu akcií na základě faktorů, jako jsou například finanční ukazatele společnosti, makroekonomické ukazatele a podobně.
- ***Klasifikace:*** Typ strojového učení, kde model je vyškolen na základě trénovacích dat a pak použit k přiřazení nových dat do kategorií na základě naučeného vzoru. Cílem klasifikace je určit, do jaké kategorie náleží daný vstup. Klasifikace se často používá v oblastech jako je rozpoznávání obrazu, analýza sentimentu, rozpoznávání řeči a další.
- ***Optimalizace:*** Jedná se o matematický přístup, který se používá k nalezení nejlepšího řešení pro daný problém.
- ***Trénink:*** Proces, pomocí kterého se model učí z tréninkových dat.
- ***Trénovací data:*** Množina dat, která se používá při trénování strojového učení. Tato data slouží jako vstup do algoritmu a pomáhají modelu se učit a určovat vztahy mezi vstupy a očekávanými výstupy. Po projití trénovacích dat, je model testován na nových datech, aby se zkontrolovala jeho přesnost a schopnost generalizace.
- ***Testovací data:*** Sada dat, která se používá k ověření správnosti a účinnosti modelu strojového učení. Tato data nejsou součástí trénovacích dat, která se používají ke zlepšování modelu, ale jsou použita k ověření, jak dobře se model dokáže vyrovnat s novými, neznámými daty. Výsledky testování pomáhají určit, jakým

způsobem by měl být model vylepšován, aby byl schopen výkonnějšího rozpoznávání a predikce.

- ***Test:*** Proces, pomocí kterého se model ověřuje na testovacích datech.
- ***Učení:*** Proces, pomocí kterého se model učí z dat a zlepšuje svůj výkon.
- ***Strojové učení:*** Interdisciplinární oblast informatiky a statistiky, zabývající se vývojem a používáním algoritmů, které se dokážou učit na datech, a tím řešit specifické úlohy bez explicitního programování. Tyto algoritmy tvoří modely na základě vzorců nalezených v datech a poté tuto naučenou informaci využívají k předpovědím v budoucnu. Strojové učení se používá v mnoha oblastech, jako například v diagnostice nemocí, řízení dodávek, analýze sentimentu a dalších.
- ***Hloubkové učení:*** Technika strojového učení, která využívá hluboké neuronové sítě k naučení funkce na základě zpracovávání dat. Tyto sítě se skládají z více vrstev, které se postupně rozšiřují a zpracovávají vstupní data, až se dostanou k výstupní vrstvě, kde jsou výstupy přiřazeny do tříd. Hloubkové učení se vyznačuje velkým počtem parametrů a schopností učit se složité funkce, což umožňuje vynikajícího výkonu v mnoha oblastech, jako je klasifikace obrázků, řešení problémů převodu jazyka a detekce odchylek.
- ***Reinforcement learning:*** Další oblast strojového učení, která se zabývá učením agentů v prostředí pomocí systému odměn a trestů. Agent se učí tak, že interaguje s prostředím a dělá rozhodnutí, která mu umožňují dosáhnout cíle. Tato rozhodnutí se hodnotí a agent je buď pochválen, nebo potrestán. Dosažené hodnoty se používají k vylepšení budoucího chování agenta, neboť ten se snaží dosáhnout nejlépe hodnocených výstupů. RL je často používán v problémech, jako jsou robotika, hraní her a optimalizace řízení.
- ***Model:*** Matematický nebo statistický popis dat nebo procesu.
- ***Agent:*** Jedná se o počítačový program nebo systém, který se učí a přijímá rozhodnutí na základě interakce s prostředím, ve kterém se nachází. Agent je schopen provádět různé akce a získávat zpětnou vazbu za svá rozhodnutí v podobě odměny nebo trestu. Cílem agenta v rámci reinforced learningu je optimalizovat své chování tak, aby maximalizoval očekávanou celkovou odměnu. Obvykle se skládají z těchto částí:

- Návrh rozhodovací strategie – agent musí vyvinout strategii pro výběr akcí v reakci na pozorování prostředí.
- Vykonání akce – agent musí být schopen provádět akce v prostředí na základě své strategie.
- Pozorování stavu prostředí – agent musí být schopen sledovat a interpretovat informace o stavu prostředí, aby mohl vyvinout vhodnou strategii.
- Zpětná vazba – agent musí být schopen interpretovat zpětnou vazbu, kterou dostává od prostředí v podobě odměny nebo trestu, aby mohl optimalizovat své rozhodování.

Použití agentů v reinforced learningu bývá aplikováno v oblastech, jako jsou hry, robotika a automatizace. Agenti se, díky tomu, že bývají trénování v určitém prostředí, učí a vyvíjí tak optimální strategie pro dané prostředí a úlohy.

- ***Generalizace:*** Schopnost modelu používat znalosti získané z tréninkových dat k úspěšnému řešení nových úkolů.
- ***Overfitting:*** Je situace, ke které dochází, když se model příliš silně přizpůsobuje konkrétním datům trénovacího množství a nezvládá dobře generalizaci nových dat. Model tedy dobře predikuje výstupy pro trénovací data, ale nová, dosud neviděná data, predikuje špatně. Jedná se o častý problém u modelů, které mají mnoho parametrů a mohou se přizpůsobit jakýmkoli vzorům v trénovacích datech.
- ***Kognitivní model:*** Matematický model, který simuluje funkce a mechanismy lidského myšlení. Tyto modely jsou často používány nejen při výzkumu kognitivní psychologie a neurověd, ale také při tvorbě kognitivních technologií, jako jsou například chatboty nebo virtuální asistenti. Cílem je zlepšovat porozumění lidské mysli, zkoumat fungování lidského myšlení a poté tyto poznatky zužitkovat při vývoji lepších technologií.
- ***Expertní systém:*** Informační systém simulující schopnosti odborníka v nějaké oblasti. Tyto systémy se často používají k řešení složitých úloh a k rozhodování v situacích, kde je potřeba kombinovat velké množství informací a znalostí. Expertní systémy se skládají z řady modulů, jako jsou *knowledge base* (databáze znalostí), *inference engine* (stroj pro dedukci) a rozhraní pro uživatele. Tyto moduly spolupracují, aby poskytly odpověď na dotazy a řešení problémů.

- ***Datové skupiny:*** Soubory dat, které jsou tříděny nebo organizovány podle určitého kritéria. Mohou se používat k různým účelům jako je analýza dat, školení modelů strojového učení, klasifikace, detekce anomálii a další. Výběr a správná příprava datových skupin mohou mít zásadní vliv na výsledky řešení. Proto je důležité zvolit správné datové skupiny a vybrat ty, které nejlépe vystihují problém, který se snažíme řešit.
- ***Neurónové sítě:*** Tyto sítě jsou matematické modely, které napodobují funkce lidského mozku a jsou často používány v hloubkovém učení.
- ***Konvoluční neurónové sítě:*** Typ hloubkových neurálních sítí, které se významně používají pro rozpoznávání obrazů a videa. Tyto sítě se skládají z konvolučních vrstev, které zachycují spatiotemporální vztahy v datech, a z plně připojených vrstev, které zpracovávají výstup z konvolučních vrstev. Mají schopnost zjednodušit a zlepšit rozpoznávání obrazů a videí.
- ***Rekurentní neurónové sítě (RNN):*** Typ sítě, který se vyznačuje schopností "pamatovat si" informace v průběhu času. Tyto sítě jsou schopné zpracovávat sériové dlouhé sekvence dat, jako jsou např. texty, hlasové signály, finanční či meteorologická data. Vyznačují se schopností uchovávat informaci o vstupu, který byl předán v minulosti a aplikovat ji na aktuální vstup. Tyto sítě jsou také schopné získávat informace o vzorcích a trendech v dlouhých sekvencích, což umožňuje použití v oblastech jako překlad jazyka, sentiment analýza, rozpoznávání řeči a další.
- ***Natural language processing (NLP):*** Jedná se o interdisciplinární obor, který se zabývá zpracováním a analýzou přirozeného jazyka. Cílem NLP je naprogramovat počítače tak, aby dokázaly pochopit, interpretovat a generovat přirozený jazyk, jako je čeština, angličtina, španělština atd. NLP je důležité pro aplikace, jako je zpracování řeči, překlad jazyků, chatboty, vyhledávání informací, sentiment analýza atd. Tyto aplikace využívají NLP k rozpoznávání přirozeného jazyka a k transformaci textových dat na informace, které mohou být následně analýzou nebo jinými postupy zpracovány. (50 AI Terms Every Beginner Should Know, 2023) (A Primer for understanding Reinforcement Learning, 2023)

Vzhledem k dynamice rozvoje umělé inteligence jsou tyto pojmy naprostým základem a pomáhají srozumitelně popisovat a pochopit různé aspekty tohoto oboru.

2 KYBERNETICKÁ BEZPEČNOST

Kybernetická bezpečnost se stala jedním z nejdůležitějších témat moderního světa. S nárůstem digitálního prostředí se zvýšil také počet kybernetických hrozeb, které mohou způsobit škody nejen pro jednotlivce, ale i pro společnosti a státy. Tato kapitola se bude zabývat tím, co je kybernetická bezpečnost, jaké jsou její hlavní hrozby a jak se proti nim bránit. Dle Výkladového slovníku kybernetické bezpečnosti (2015) je kybernetická bezpečnost: *„Souhrn právních, organizačních, technických a vzdělávacích prostředků směřujících k zajištění ochrany kybernetického prostoru.“*

Týká se tedy ochrany počítačových systémů, sítí a dat před neoprávněným přístupem, poškozením nebo krádeží. Cílem kybernetické bezpečnosti je zajistit ochranu informací a dat, a tím chránit jednotlivce, společnosti a státy.

2.1 Hlavní hrozby kybernetické bezpečnosti

Kybernetické hrozby mohou být různého zdroje a typu.

Dělení dle zdroje hrozeb:

- Hrozby způsobené člověkem.
- Technické chyby.
- Vyšší moc. (Kolouch a Bašta, 2019)

Mezi **hrozby způsobené člověkem** nejčastější patří:

- ***Malware*** – Je jeden z nejrozšířenějších typů kybernetických hrozeb. Jedná se o software, který je navržen tak, aby napadl počítač nebo jiné zařízení a způsobil škodu nebo ukradl citlivá data. Mezi typy malware patří viry, červi, trojské koně a ransomware.
- ***Phishing*** – Kybernetická hrozba, která se obvykle vyskytuje v e-mailech. Útočníci se vydávají za legitimní osoby nebo organizace a snaží se získat citlivá data od obětí. To může zahrnovat hesla, kreditní karty nebo jiná osobní data.
- ***DDoS útoky*** – Distribuované útoky odmítnutí služby (DDoS) jsou útoky, při kterých se velké množství počítačů snaží přetížit server nebo síť a znemožnit tak přístup k internetovým službám.

- ***Insider hrozby*** – Jedná se o kybernetické útoky, které jsou spáchány v rámci organizace. Jedná se o zaměstnance, kteří zneužívají svých oprávnění nebo přístupových práv k neoprávněnému přístupu k datům nebo k vytvoření hrozeb pro organizaci.
- ***Botnets*** – Jsou sítě počítačů, které jsou infikovány malwarem a které mohou být použity k vytvoření DDoS útoků nebo k ukradení dat.

Kromě těchto nejčastějších druhů hrozeb existuje mnoho dalších, jako jsou například útoky na bezdrátové sítě, phishing útoky přes sociální média a mnoho dalších. Je důležité, aby organizace a jednotlivci byli obezřetní a měli k dispozici adekvátní nástroje a postupy k ochraně před těmito hrozbami.

Technické hrozby lze rozdělit na chyby softwaru a hardwaru.

- ***Chyby softwaru*** – Mohou být příčinou nesprávného fungování aplikace či nepatřičného chování systému. V jejich důsledku se také může stát, že systém produkuje nesprávné výsledky či produkty. Příčin může být celá řada. Od nedoladěné používané verze SW, přes chybu způsobenou zásahy do struktury SW, nesprávným používáním či snahou aplikovat SW v prostředí, pro které nebyl zamýšlen. Velká část případů chyb SW, ale bývá způsobená cyklením chyb a ignorací ze strany obsluhy zařízení, nejen samotným selháním SW. Stejně jako u chyb HW, ani zde nelze ze seznamu vyškrtnout zásah zvenčí.
- ***Chyba hardware*** – V dnešní době se, vzhledem k trendu minimalizace a optimalizace, u HW součástí minimalizuje výskyt pohyblivých částí HW zařízení. I tak se ovšem jedná o zařízení s určitou konstrukcí, výkonem a životním cyklem, která mohou selhat. Nemluvě o tom, že může dojít k narušení zabezpečení a může dojít k jejich selhání vnějším zaviněním, jak dokládají například úspěšně izraelské útoky na iránská zařízení.

Hrozby v podobě **vyšší moci**, zejména v podobě výpadku proudu či např. požáru nezpůsobeného lidským přičiněním, jsou do určité míry pokrytá pomocí zařízení UPS a zálohováním dat, protože hrozby přesahující tyto možnosti měly být zapracovány v průběhu analýzy rizik subjektu a dle toho ošetřeny.

2.2 Obrana proti kybernetickým hrozbám

Existuje mnoho technologií a nástrojů, které lze použít k ochraně před kybernetickými hrozbami. Některé z těchto nástrojů jsou:

- ***Firewall*** – Je ochranný prvek, který může být implementován na síťové úrovni a chrání před neoprávněným přístupem k počítačovým systémům. Firewall umožňuje kontrolu přístupu k síti a může blokovat nebezpečné připojení.
- ***Antivirový software*** – Pomáhá chránit počítače a další zařízení před malwarem a dalšími škodlivými programy. Antivirové programy běžně kontrolují všechny soubory na počítači a hledají podezřelé aktivity.
- ***Šifrování dat*** – Umožňuje chránit citlivá data před útoky a zajišťuje, že tyto údaje nejsou snadno přístupné neoprávněným osobám. Při šifrování dat jsou data převedena do kódu, který je nečitelný bez správného klíče.
- ***Multifaktorové ověřování*** – Vyžaduje více než jednu formu ověření identity uživatele, aby se předešlo neoprávněnému přístupu k datům a systémům. Tento proces zahrnuje běžné faktory, jako jsou hesla, spolu s dalšími faktory, jako jsou biometrická data nebo SMS kódy.
- ***Pravidelné zálohování dat*** – Je důležité pro obnovu dat v případě, že dojde k narušení nebo ztrátě dat. Zálohování dat zajišťuje, že data jsou k dispozici i v případě, že jsou poškozena nebo ztracena. (Kolouch a Bašta, 2019)
- ***Intrusion Detection System (IDS)*** – jsou bezpečnostní nástroje, které pomáhají chránit počítačové sítě před útoky a neoprávněným přístupem. Tyto systémy sledují provoz na síti a hledají podezřelé aktivity, které mohou signalizovat pokus o narušení bezpečnosti.

  Existují dva hlavní typy IDS systémů: síťové a hostované IDS. Síťové IDS sledují provoz na síťové úrovni a mohou být umístěny na hranicích sítě, jako jsou například firewally. Hostované IDS sledují aktivitu na samotných počítačích a mohou být umístěny na jednotlivých zařízeních v síti.

  IDS systémy mohou fungovat buď v režimu detekce, nebo v režimu prevence. V režimu detekce IDS systémy pouze sledují provoz a hledají podezřelé aktivity, ale nezasahují do provozu. V režimu prevence IDS systémy mohou blokovat nebezpečné připojení, aby zabránily útokům.

IDS systémy mohou být konfigurovány k detekci různých typů útoků, jako jsou například odmítnutí služby (DoS), útoky na zabezpečení webových aplikací, útoky na síťové protokoly a další. Tyto systémy používají různé metody detekce, včetně srovnávání provozu s pravidly a heuristické analýzy.

IDS systémy jsou důležitým prvkem kybernetické obrany a mohou pomoci organizacím předejít útokům a minimalizovat rizika kybernetických hrozeb. (Intrusion Detection System, 2023)

- ***Intrusion Prevention System (IPS)*** – systémy jsou další vrstvou kybernetické obrany, která se podobá IDS systémům, ale s tím rozdílem, že IPS systémy nejen detekují, ale také přímo blokují útoky na počítačové sítě.

  IPS systémy pracují v reálném čase a provádějí aktivní analýzu provozu v síti. Pokud detekují útok, systém ihned blokuje podezřelý provoz a uzavře přístup pro potenciálního útočníka. IPS systémy používají různé techniky k identifikaci a blokování nebezpečných aktivit, jako je například filtrování provozu, blokování přístupu k určitým webovým stránkám nebo například detekce a blokování pokusů o prolomení hesla.

  Jako IDS systémy, IPS systémy mohou být umístěny v různých částech počítačové sítě, včetně brány, síťových prvků nebo na koncových zařízeních.

  Hlavním cílem IPS systémů je zajistit rychlou reakci na útok a zabránit poškození sítě a zařízení, přičemž minimalizují přerušení služeb. V kombinaci s ostatními bezpečnostními technologiemi jako jsou firewally, antivirové programy nebo IDS systémy mohou IPS systémy poskytnout komplexní a účinnou ochranu počítačové sítě proti různým typům kybernetických hrozeb. (Intrusion Prevention System, 2023)

- ***Security Information and Event Management (SIEM)*** – jsou bezpečnostní nástroje používané organizacemi pro sběr, analýzu a sledování bezpečnostních událostí v počítačových sítích a systémech. Tyto systémy umožňují centralizované řízení bezpečnostních událostí v organizaci a poskytují rychlou a efektivní odezvu na bezpečnostní hrozby.

  SIEM systémy sbírají data z různých zdrojů, včetně bezpečnostních logů, sítě, systémových a aplikativních událostí. Tyto data jsou následně analyzována a vyhodnocována za účelem identifikace potenciálních bezpečnostních hrozeb a anomálií v počítačových systémech. Výsledky analýzy jsou následně prezentovány

v přehledném dashboardu nebo výstupech, které umožňují administrátorům rychlou reakci na bezpečnostní incidenty. SIEM systémy také umožňují vytváření pravidel pro detekci specifických bezpečnostních hrozeb a upozornění administrátorů, když se tyto hrozby objeví. SIEM systémy jsou důležitým nástrojem pro zajištění bezpečnosti počítačových sítí a ochranu před útoky ze strany hackerů a jiných škodlivých entit.

Kromě technických řešení jsou také zásadní školení zaměstnanců, kteří jsou první linií obrany proti kybernetickým útokům. Důležitým krokem je také vytvoření bezpečnostní politiky a standardů, které stanoví způsoby ochrany počítačových systémů, dat a jak se s nimi nakládá. Dalším způsobem, jak se chránit proti kybernetickým hrozbám, je vzdělávání a osvěta veřejnosti o nebezpečích a způsobech ochrany. (Kolouch a Bašta, 2019)

3 UMĚLÁ INTELIGENCE

Umělá inteligence (AI) je odvětví informatiky zaměřené na vývoj algoritmů a systémů, které jsou schopny vykonávat úkoly, které by normálně vyžadovaly lidskou inteligenci, jako je učení, rozpoznávání řeči, rozumění textu, rozpoznávání obrazů, plánování a rozhodování. Cílem AI je vytvořit počítače, které budou schopny fyzických i mentálních činností, jako lidé.

3.1 Rozdělení a druhy umělé inteligence

Existuje několik způsobů, jak lze rozdělit umělou inteligenci. Mezi hlavní kategorie patří:

- ***Podle úrovně inteligence:*** Umělá inteligence může být klasifikována jako *nízkoúrovňová* (např. automatická kontrola a řízení) nebo *vysokoúrovňová* (např. strojové učení a počítačový vidění).
- ***Podle schopnosti učení:*** Umělá inteligence může být klasifikována jako *statická* (např. klasické programování) nebo *adaptivní* (např. strojové učení).
- ***Podle úkolu:*** Umělá inteligence může být klasifikována jako *specifická* (např. rozpoznávání řeči) nebo *univerzální* (např. GPT-3).
- ***Podle činnosti:*** Umělá inteligence může být klasifikována jako *reaktivní* (např. hry) nebo *proaktivní* (např. plánování).
- ***Podle zdroje inspirace:*** Umělá inteligence může být klasifikována jako *biologická* (např. neuronové sítě) nebo *nebiologická* (např. symbolické systémy).

Následně lze AI zařadit k jednotlivým generacím a druhům, mezi které patří:

- ***Umělá inteligence první generace:*** Tato AI se zaměřuje na přímé naprogramování počítače, aby vykonával specifické úkoly.
- ***Umělá inteligence druhé generace:*** Tato AI se zaměřuje na strojové učení, což umožňuje počítačům, aby se učily z dat a vykonávaly úkoly bez přímého naprogramování.
- ***Umělá inteligence třetí generace:*** Tato AI se zaměřuje na rozumění a přirozené komunikaci, což umožňuje počítačům, aby se učily z dat a vykonávaly úkoly bez přímého naprogramování a rozuměly jim.

- ***Umělá inteligence čtvrté generace:*** Tato AI se zaměřuje na vytvoření umělé inteligence, která je schopna se učit jako člověk a má schopnosti jako člověk.
- ***Umělá inteligence v kognitivním stylu:*** Tato AI se zaměřuje na imitování kognitivních procesů lidského mozku, jako jsou vnímání, myšlení, rozpoznávání a učení.
- ***Umělá inteligence v komponentním stylu:*** Tato AI se skládá z různých komponent, které spolu komunikují a spolupracují při řešení úkolů. (Russell a Norvig, 2010)

3.2 Milníky ve vývoji umělé inteligence

Obor AI se, ve své podstatě, vyvíjí již od starověku, neboť už řečtí filozofové se zabývali otázkou lidské inteligence a jejím napodobením či nahrazením. Opravdový rozmach ale přišel až s počítači. Zprvu to sice vázlo na nedostatečné výpočetní kapacitě, ale s rozvojem celého odvětví došlo i k rozvoji AI. Některé z hlavních milníků zahrnují:

- 1936 - První teorie výpočetního výkonu: Publikoval ji Alan Turing v jeho článku "On Computable Numbers, with an Application to the Entscheidungsproblem". Tuto teorii lze považovat za základ teoretického počítačového vědění a stala se klíčovou pro rozvoj počítačů a informačních technologií.
- 1954 - První strojový překlad: Josephem Weiserem a Warrenem McCullochem vytvořili první strojový překladový program.
- 1956 - Dartmouth Conference: Pořádali ji John McCarthy, Marvin Minsky, Nathaniel Rochester a Claude Shannon. Poprvé se zde použil termín „umělá inteligence“ a poprvé se zde definovala jako věda o počítačové inteligenci.
- 1966 – Započatí prvního AI projektu v rámci MIT.
- 1969 – Vynález LISP programovacího jazyka.
- 1974 – První použití expertního systému.
- 1979 – Započatí prvního projektu "Expertních systémů".
- 1986 – První neuronová síť: Geoffrey Hinton a jeho tým oznámili vytvoření první hloubkové neuronové síť, což bylo důležité pro rozvoj hloubkového učení.
- 1987 – První použití rekurentních neuronových sítí.

- 1988 – Kohonen přišel s konceptem samoorganizujících se map a využil je k vyhodnocení lidské řeči.
- 1997 – Výhrou superpočítače Deep Blue nad šachovým mistrem Garrym Kasparovem se začal rozvíjet deep learning.

Obr. 1 Kasparov během své 4. hry proti IBM Deep Blue (Zdroj: 20 Years after Deep Blue: How AI Has Advanced Since Conquering Chess, 2023)

- 2006 – První úspěšné použití konvolučních neuronových sítí.
- 2010 – Objev generativních adversárních sítí.
- 2015 – První úspěšné použití deep reinforcement learningu. (Russell a Norvig, 2010)

Z těchto milníků stojí za zmínku zejména jednotlivé teorie:

Rumelhartova koncepce několikavrstvých sítí se spojitým chováním (anglicky "*Rumelhart's concept of multilayer neural networks with continuous behavior*") je model neuronových sítí, který umožňuje učení se vícerozměrných vzorů a vztahů mezi nimi. Tento model se skládá z několika vrstev neuronů, přičemž každá vrstva přijímá vstupní signály a generuje výstupní signály pro následující vrstvu. Každý neuron v síti je spojen s neurony v předchozí i následující vrstvě pomocí vah. Tyto váhy jsou inicializovány náhodně a poté

se postupně upravují během učení sítě. Během učení se snaží síť minimalizovat chybu výstupu v porovnání s očekávaným výstupem, a to pomocí algoritmu zpětného šíření chyby („*backpropagation*“).

Také umožňuje využití aktivace spojitého typu v neuronových sítích, což znamená, že vstupní signály i výstupní signály jsou spojité funkce, nikoli binární. Díky tomu je možné pracovat s velkým množstvím vstupů a výstupů, což umožňuje modelování složitých datových struktur. Tento model umožňuje také řešení problémů jako například klasifikace a rozpoznávání vzorů, zpracování obrazu, zpracování řeči, překladu přirozeného jazyka a mnoho dalších. Rumelhartova koncepce několikavrstvých sítí se spojitým chováním je jedním z nejúspěšnějších modelů neuronových sítí a zůstává významnou součástí strojového učení.

Kohonenovy samoorganizující se mapy (anglicky "*Kohonen self–organizing maps*", zkráceně SOM) jsou neuronové sítě, které se používají pro vizualizaci a analýzu vícerozměrných dat. Tyto mapy využívají nepřeberné množství vstupních dat a uspořádávají je do dvourozměrného (nebo třírozměrného) prostoru, což umožňuje vizualizaci datových vzorců a zjištění jejich vzájemných vztahů. SOM se skládá z vrstvy neuronů, kde každý neuron představuje jeden bod v mapě. Každý neuron má své váhy, které určují jeho pozici v prostoru. Váhy neuronů jsou inicializovány náhodně a poté se postupně upravují během trénování sítě. Trénování sítě začíná prezentací vstupních dat. Každé vstupní datum je představováno jako vektor a každý neuron v síti je ohodnocen podle toho, jak dobře se jeho váhy shodují s vektorem vstupních dat. Neuron s nejlepším ohodnocením je označen jako "vítěz" a jeho váhy jsou dále upraveny tak, aby byly bližší k vektoru vstupních dat. Součástí tohoto procesu je také úprava vah sousedních neuronů – všechny neurony v blízkosti vítězného neuronu se upraví podobným způsobem. Následně neurony, které reprezentují podobná vstupní data, jsou přitahovány k sobě a tvoří shluky.

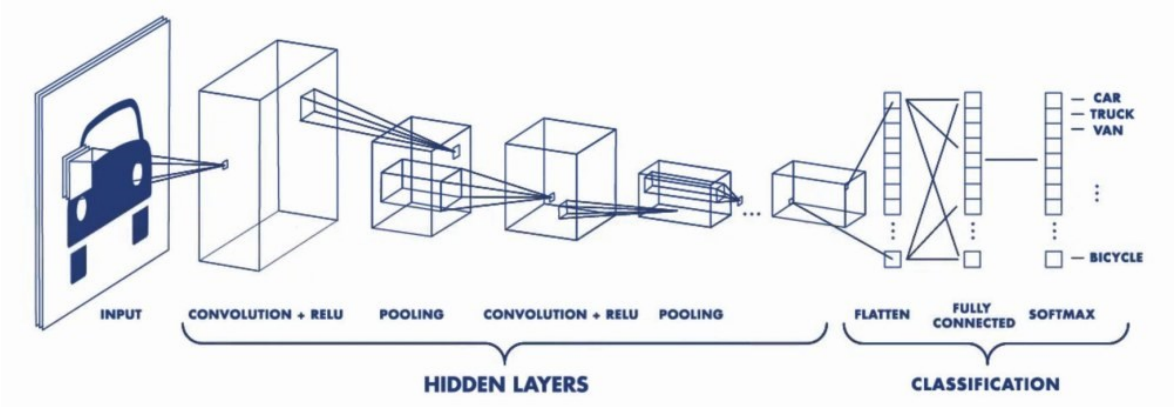
Po trénování sítě se vstupní data zobrazí na dvourozměrné mapě, kde každý neuron představuje jeden bod. Shluky neuronů na mapě reprezentují shluky podobných vstupních dat. SOM má mnoho aplikací, včetně analýzy obrazů, zpracování řeči, analýzy textu, klasifikace dat, vizualizace dat a mnoho dalších. SOM umožňují efektivní shlukování dat, zjišťování jejich vzájemných vztahů a vizualizaci složitých datových struktur.

Neuronové sítě s radiální bází (RBF) jsou typem neuronové sítě, která se používá pro aproximaci funkcí a klasifikaci dat. Tato síť se skládá z několika vrstev neuronů, kde každý

neuron v první vrstvě se nazývá radiální bázová funkce (RBF). Radiální bázové funkce jsou funkce, které mají vysokou hodnotu pro vstupy blízké určitému středu a klesají rychle k nule pro vstupy vzdálené od středu. Tyto funkce slouží k rozpoznání a reprezentaci vzorů vstupních dat. Každý neuron v první vrstvě sítě používá jednu radiální bázovou funkci, která je centrována v nějakém bodě vstupního prostoru. Druhá vrstva sítě je lineární vrstva, která slouží k vážení vstupů z radiálních bázových funkcí a vytváření výstupu sítě. Váhy v této vrstvě jsou určeny během trénování sítě pomocí algoritmu učení. Proces trénování RBF sítě probíhá tak, že jsou nejprve určeny polohy středů radiálních bázových funkcí a škálovací faktor, který určuje, jak rychle se hodnota radiální bázové funkce klesá s rostoucí vzdáleností od středu. Poté jsou určeny váhy v druhé vrstvě, aby síť mohla co nejlépe aproximovat výstup pro trénovací data. RBF sítě mají několik výhod. Jsou efektivní pro rychlé učení a pro aproximaci funkcí s vysokou přesností. Jsou také velmi vhodné pro klasifikaci dat, kdy je vstupní prostor vysokorozměrný, protože radiální bázové funkce umožňují efektivní reprezentaci vstupních dat. Nicméně, RBF sítě mají vysokou složitost a výpočetní náročnost pro trénování, což může být nevýhoda pro velké sítě nebo pro data s velkým počtem dimenzí.

Konvoluční neuronové sítě (CNN) jsou druhem neuronových sítí, které se často používají pro zpracování obrazových a zvukových dat. Tato síť využívá konvoluci jako hlavní operaci pro extrahování rysů ze vstupních dat. V konvoluční neuronové síti jsou vstupní data zpravidla obrazové matice, které jsou předány konvoluční vrstvě sítě. Konvoluční vrstva obsahuje filtry (nebo jádra), které se skládají z vah. Tyto filtry jsou aplikovány na vstupní data pomocí konvoluce, což je matematická operace, která vynásobí jednu matici (v tomto případě vstupní data) druhou (filtr) a sečte výsledky. Každý filtr v konvoluční vrstvě detekuje určité rysy v datech, jako jsou hrany, textury, tvary atd. Konvoluční vrstva vytváří mapy příznaků, které zobrazují jak daný rys (detekovaný filtrem) prochází vstupními daty. Tyto mapy příznaků jsou poté předány další vrstvě sítě pro další extrakci rysů. Další vrstvy sítě se skládají z poolingové vrstvy[1] a plně propojené vrstvy[2]. Konvoluční neuronové sítě jsou velmi úspěšné v úkolech, jako je klasifikace obrazů, rozpoznávání objektů v reálném čase, segmentace obrazu a další. Tyto sítě jsou velmi efektivní v učení z velkých datových sad a při použití grafických karet mohou být trénovány rychle.

[1] Snižuje rozměry mapy příznaků tím, že vybírá pouze nejdůležitější informace.
[2] Převádějí mapy příznaků na konečný výstup.



Obr. 2 Popis funkce konvoluční neuronové sítě. (Zdroj: Deep learning pro segmentaci obrazu, 2023)

- ***Generativní adversární sítě (GAN)*** jsou typem neuronových sítí, které jsou schopny generovat nová data, například obrazy nebo zvuky. GAN se skládá ze dvou hlavních částí – generátoru a diskriminátoru, které spolu soupeří v učení. Generátor vytváří nová data, zatímco diskriminátor posuzuje, zda jsou data skutečná nebo vygenerovaná. Během trénování se generátor snaží produkovat data, která diskriminátor nepozná od skutečných dat. Tímto způsobem se generátor učí, jak vytvářet autentická data, zatímco diskriminátor se učí rozpoznávat skutečná data od vygenerovaných. Cílem GAN je naučit generátor vytvářet taková data, která jsou realistická a neodlišitelná od dat ve vstupním datasetu. GAN může být použit například k tvorbě fotorealistických obrázků, syntetizaci hlasů, vylepšování kvality obrazu, překladu jazyků a mnoha dalších aplikací. Přestože GAN může produkovat velmi realistická data, je trénování GAN náročné a může trvat mnoho hodin nebo dokonce dnů. Kromě toho může GAN trpět problémem nedostatečné diverzity, kdy generátor produkuje pouze několik typů dat a opakuje je, místo aby vytvářel různorodá data. (Goodfellow, Bengio a Courville, 2016) (Neuronové sítě - Elements of AI, 2023) (Úvod do neuronových sítí, 2005) (What's the difference between CNN and RNN?, 2023)

Tyto milníky jsou jen několika příklady toho, jak se umělá inteligence vyvíjela a jaký vliv měla na svět vědy a technologie. V současné době se umělá inteligence dále rozvíjí a její využití se šíří do mnoha odvětví, jako je například zdravotnictví, finančnictví a automobilový průmysl.

3.3 Využití umělé inteligence v kybernetickém zabezpečení subjektu

Umělá inteligence může hrát důležitou roli v kybernetickém zabezpečení subjektu. Některé z možností využití AI v kybernetickém zabezpečení zahrnují:

- ***Detekce hrozeb:*** S pomocí AI lze analyzovat velké množství dat a identifikovat podezřelé aktivity nebo chování, které by mohly naznačovat hrozbu pro subjekt. Toto je obzvláště užitečné při detekci pokročilých hrozeb, které se snaží zůstat skryty.
- ***Zajištění bezpečnosti sítě:*** AI může být použita ke sledování sítě a detekci anomálií, což umožňuje rychlou reakci na hrozby a minimalizaci škod.
- ***Identifikace zranitelností:*** Pomocí AI lze identifikovat zranitelnosti v systémech, aplikacích a infrastruktuře subjektu a tak snížit pravděpodobnost úspěšného útoku.
- ***Prevence podvodů:*** AI může být použita k detekci podvodů, jako jsou phishingové e-maily, a tak snížit riziko úspěšného útoku na subjekt.
- ***Automatická odezva:*** S využitím AI lze vyvinout automatické mechanismy reakce na hrozby, např. uzavření přístupových cest, blokování útočníků nebo šifrování dat.
- ***Zajištění bezpečnosti aplikací:*** AI může být použita k identifikaci nebezpečných kódů a chování v aplikacích, což umožňuje zvýšení bezpečnosti celého systému.
- ***Předpovídání hrozeb:*** S využitím AI lze provádět analýzu dat a předpovídat potenciální hrozby, což umožňuje přijmout preventivní opatření a minimalizovat riziko útoku. (Kolouch a Bašta, 2019)

Využití umělé inteligence v kybernetickém zabezpečení je tedy velmi rozmanité a může zlepšit bezpečnost subjektu v mnoha různých oblastech. Tato práce se ale zaměří na využití AI k zefektivnění fungování video dohledového systému (VDS) subjektu. Zúžení tématu na video dohledové systémy vychází z toho, že v současné době jsou video dohledové systémy využívány v mnoha oblastech, od průmyslu a obchodu po veřejnou dopravu a městskou infrastrukturu. Tyto systémy často slouží k monitorování a řešení bezpečnostních situací, jako jsou krádeže, vandalství a narušení veřejného pořádku.

Ruku v ruce s jejich rozšířením jde také nárůst kybernetických hrozeb, které se týkají VDS, neboť kamery využívané v menších provozech či domácnostech jsou, stejně jako jiná IoT zařízení, často vystavena různým druhům kybernetických hrozeb. Tyto hrozby mohou zastupovat například útoky na systémy a sítě, úniky dat a využívání zranitelností v kódu

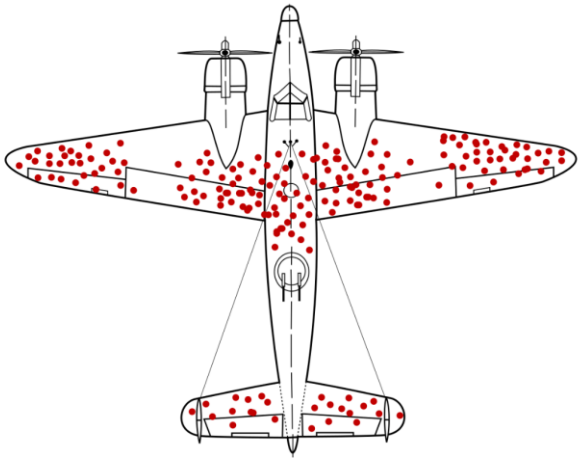
a software. Právě proto by zaměření se na tuto oblast mohlo přinést cenné poznatky o tom, jak se tyto systémy mohou lépe chránit před kybernetickými útoky a jak mohou být zabezpečeny tak, aby byly spolehlivé a efektivní v plnění svých úkolů. V rámci této práce budou navrženy a vyzkoušeny různé metody a algoritmy, které by mohly pomoci zvýšit bezpečnost VDS. Zaměření se na bezpečnost připojení jednotlivých součástí VDS by tedy mohlo poskytnout ucelenější pohled na celkovou bezpečnost těchto systémů a jedním z cílů této práce je i zamyslet se nad vývojem nových řešení v oblasti kybernetické bezpečnosti VDS. V této oblasti lze očekávat největší přínos AI v oblastech:

- ***Detekce anomálií:*** AI může být použita k detekci anomálií v chování uživatelů, aplikací a zařízení v sítích subjektu. Toto umožňuje rychlou reakci na možné hrozby a minimalizaci škod.
- ***Identifikace zranitelností:*** AI může být použita k identifikaci zranitelností v systémech a infrastruktuře subjektu. To umožňuje subjektu přijmout preventivní opatření a minimalizovat riziko úspěšného útoku.
- ***Automatická odezva:*** S využitím AI lze vyvinout automatické mechanismy reakce na hrozby. Například AI může být použita k uzavření přístupových cest, blokování útočníků nebo šifrování dat. V této oblasti AI vykazuje lepší výsledky než algoritmy založené na strojovém učení.
- ***Předpovídání hrozeb:*** S využitím AI lze provádět analýzu dat a předpovídat potenciální hrozby. To umožňuje přijmout preventivní opatření a minimalizovat riziko útoku.
- ***Monitorování bezpečnostních kamer:*** AI může být použita k analýze videa z bezpečnostních kamer, což umožňuje identifikaci podezřelých aktivit a chování díky pokročilým možnostem vyhodnocování obrazu. (Russell a Norvig, 2010)

Z výše uvedeného vyplývá, že pomocí AI vhodně zakomponované do VDS, lze dosáhnout zabezpečení, které bude mít velmi málo, pokud vůbec nějaké, slepé místo. Takový systém by sestával z:

- ***Kamer:*** Ty, řízené AI, by byly rozdělené dle střežených zón. Zóny by měly danou prioritu a důležitost. Dle toho by se jednak v nich sledovalo dění, jednak by se proměnlivě vyčleňoval výpočetní výkon. Nejnižší prioritu by mělo dění za vnější hranicí střeženého subjektu, zejména kvůli možnému zahlcení vyhodnocovacích algoritmů způsobených provozem. Priorita zón by se postupně zvyšovala. Od vnitřního areálu subjektu, přes vstupy a vjezdy do jednotlivých úseků, až po střežené oblasti s omezeným přístupem jako serverovna, strojovna nebo uložiště cenných informací a materiálů.

  U tohoto rozdělení kamer a jejich priorit by bylo vhodné nezapomenout na princip „*Survivorship Bias*“ (v překladu klam přeživších). V kontextu dohledu se jedná o situaci, kdy by algoritmus strojového učení (na základě regrese) soustředil výpočetní výkon k nejvytíženějším kamerám (typicky by se mohlo jednat o časové úseky, kdy by přicházeli či odcházeli zaměstnanci do zaměstnání) a v důsledku toho mu uniklo dění v méně exponované oblasti.



Obr. 3 Pravděpodobně nejznámější příklad „klamu přeživších“ (Zdroj: Wikipedia, 2023)

Kamery u vstupů do vyhrazených zón umožňují i rozeznání předmětů, které by si vstupující osoba nesla s sebou, a mohly by upozornit obsluhu v případě, kdy by si tato soba ve střežené zóně něco nechala, případně si z ní něco odnášela.

Důležitým aspektem by u kamer, v souvislosti s kybernetickou bezpečností, bylo vyhodnocování obrazu. V ideálním případě by se tato problematika opět dělila dle

zón. Málo exponované zóny by bylo možné řešit na cloudu s nižším zabezpečením a exponované zóny by bylo možné řešit na vlastní síti subjektu, fyzicky oddělené, chráněné firewallem a VPN serverem, aby nedošlo k získání záběrů neoprávněnými osobami.

- ***Čidel:*** Čidla by zejména na perimetru subjektu pomáhala odfiltrovávat zatížení kamer. V praxi se reálně využívá nasměrování spárované bezpečnostní kamery ve směru zdroje předem určeného zvuku[3], zachyceného přidruženými mikrofony. Podobně to funguje i s rotací předurčených kamer do míst narušení střežení infračervenými závorami, pohybovými čidly a podobně. Tyto řešení umožňují snížit nároky na výpočetní výkon dohledového softwaru při zachování vysoké účinnosti systému. A také se to významnou měrou podílí na snížení rizika falešných alarmů, způsobených působením vnějších vlivů na čidla.
- ***Biometrických údajů***: V zónách s vyšší prioritou a bezpečností by kamerový systém mohl ověřovat pracovníky na základě biometrických údajů. Primárně z toho důvodu, že by se již jednalo o menší prostory a řádově menší počet osob. To by algoritmům umožňovalo ověřovat oprávněnost pobytu osob v dané zóně. V případě menších počtů osob by to také mohlo suplovat kontrolu docházky. V neposlední řadě by bylo možné toto sledování pohybu osob využít v případě MU, kdy by měly zasahující jednotky větší přehled, zda došlo k evakuaci všech osob, které se v zóně pohybovaly, nebo je ještě třeba někoho dohledat a evakuovat.
- ***RFID čipů:*** Tato technologie by vhodně doplňovala kamerový systém v zónách s menší bezpečnostní prioritou a vyšším množstvím pracovníků. Kamery by se zaměřily na přístupový terminál jen v případě neplatného čipu nebo čipu s oprávněním pro jinou oblast a informovaly by obsluhu VDS. Tyto RFID čipy by v případě vhodně zvoleného grafického provedení (například by byly barevně rozlišené dle oprávnění), mohly při viditelném nošení pomáhat kamerovému systému při detekci nepovolených či neoprávněných vstupů.
- Posledním stupněm ochrany by bylo tzv. ***RBAC (Role-Based Access Control)***. Jedná se o metodu řízení přístupu k datům a zdrojům v informačním systému na základě oprávnění, která jsou přidělena jednotlivým uživatelům nebo skupinám uživatelů. Každá skupina má určitá oprávnění a povinnosti, které určují, na jaké informace

[3] havárie, tříštící se sklo apod.

> a zdroje má uživatel v této skupině přístup. RBAC také umožňuje snadné spravování a údržbu přístupových práv, protože je založen na rolích, nikoli na jednotlivých uživatelích. (Šalomon, 2021)

Takto zvolený a vybavený systém by tedy mohl obsluze dohledového centra značně zjednodušit práci. V této sestavě je ideální kombinací využití strojového učení, které v reálném čase obraz vyhodnocuje podle předem nastavených algoritmů, a umělé inteligence, která by se zaměřila na koordinaci činnosti s dalšími prvky zabezpečovacího systému, vyhledávání anomálií a v případě jejich nalezení by spustila adekvátní odezvu. Od upozornění obsluhy VDS po uzavření objektu.

4 VIDEO DOHLEDOVÉ SYSTÉMY

Video dohledové systémy jsou ve své podstatě dalším stupněm vývoje bezpečnostních kamerových systémů. Kombinace vyspělých ovládacích algoritmů a moderních kamerových systémů lze vytvořit dohledový systém do jakýchkoliv podmínek.

4.1 Kamery

Videokamery tvoří základní prvek video dohledových systémů. První videokamery byly těžké, neskladné a na obsluhu náročné zařízení s omezenou kapacitou záznamu. Jejich využití se omezovalo na experimentální účely a zaznamenávání laboratorních i jiných pokusů. V této převážně statické roli jejich rozměry, a s ním spojený nedostatek mobility, nebyly na obtíž. Jako převážná část jiných technologií byla významným milníkem 2. světová válka. Pokrok se projevil i u kamer, které zaznamenaly zvýšený zájem obzvláště ve chvíli, kdy se začaly používat i pro jiné než vědecké účely. Poté, díky rozmachu filmového průmyslu, zaznamenaly překotný vývoj. Zvlášť významný posun zaznamenaly v druhé polovině 20. století, kdy docházelo ke zmenšování rozměrů a jejich zpřístupnění širší veřejnosti. Signifikantní roli sehrály během 80. let firmy Sony[4] a JVC[5].

Obr. 4 Porovnání bezpečnostních kamer – 80. léta a současnost (Zdroj: Deep Sentinel, 2023)

[4] s jejími produkty Betamax
[5] s řadou VHS-C

Dnes jsou kamery nabízené v široké škále velikostí a výbav určujících jejich schopnosti. Od kamer viditelně umístěných v běžném kamerovém pouzdře, jejichž účelem je prevence, přes ty v zodolněném pouzdře umístěných na exponovaném místě, kde hrozí jejich poškození, jako jsou třeba sportovní stadiony, nebezpečné provozy atd.

Kamery nicméně nemusí být vždy přiznané. I toto odvětví zasáhl trend miniaturizace a díky digitálním technologiím už není pro pořízení záznamu nutný objemný optický element. Mohou tak být zabudovány do vhodně zvoleného okolního prvku, čímž lze předejít situacím, kdy by kameře hrozilo poškození. Skryté kamery ovšem nemusí být užity jen jako prevence jejich poškození, ale také z důvodu, kdy nechceme, aby poutaly pozornost. Ve finále lze tyto skryté kamery uplatnit také ke skrytému monitorování prostor.

Dále se kamery dělí na černobíle, barevné a kombinované. Zajímavosti kombinovaných kamer je to, že reagují na hladinu světla a za zhoršených podmínek fungují jako černobílé kamery. Složitější kombinované kamery mohou používat infračervené reflektory k přísvitu, kdy toto pro lidské oko neviditelné světlo umožňuje kamerám pořizovat kvalitní záznam i za snížených světelných podmínek, což značně zvyšuje jejich využitelnost.

Pro video dohledové systémy je ovšem nejpodstatnější rozdělení kamer na analogové a digitální.

- ***Analogové:*** Použití analogových kamer sebou nese nutnost použití multiplexeru, záznamového zařízení, monitorů a rozsáhlých (samozřejmě dle počtu kamer, vzdáleností atp.) kabelových svazků. Kabelové svazky představují největší zdroj komplikací při návrhu a provozování těchto systémů. Nejen kvůli nutnosti vedení oddělených kabelů pro obraz a napájení, aby se předešlo rušení, ale také kvůli tomu, že každá další kamera dodatečně začleněná do systému vyžaduje nové kabelové vedení.
- ***IP kamery:*** Začínají v posledních letech převažovat při návrzích zabezpečovacích systémů zejména díky absenci složitých kabelových svazků, protože v případě PoE IP kamer stačí každé kameře jeden kabel až do vzdálenosti 100 m. Nemluvě o možnosti použít vnitřní zabezpečenou síť k přenosu obrazu a použít kameru s vlastním zdrojem napájení. Vedle toho IP kamery neposkytují jen jednosměrný datový tok, ale v kombinaci s různými dodávanými aplikacemi mohou plnit vícero úkolů.

Oba druhy kamer potřebují pro svou funkci médium pro ukládání záznamu k případnému pozdějšímu vyhodnocení. Dnes se již prakticky nesetkáme se starými CCTV systémy, které používaly videokazety k záznamu obrazu a fungovaly nejčastěji ve smyčce, kdy se po určité délce záznamu začal záznam nahrávat opět od začátku nosiče přes starší záznam. U analogových kamer se v současnosti používají spíše hybridní systémy, které kombinují analogové kamery a digitální uložiště. U IP kamer funguje ukládání na harddisky případně na cloud, což umožňuje uživatelům i vzdálený přístup k záznamům a jeho další zpracování.

V souvislosti s dalším zpracováním a vyhodnocením obrazu video dohledové systémy rozdělit na:

- ***Systémy s obsluhou:*** Jedná se převážně o dohledová centra využívající personál k ovládání kamer a vyhodnocování záznamu. Tyto systémy jsou ovšem náročné z hlediska zabezpečení provozu a kladou značné nároky na obsluhu.
- ***Semi-autonomní systémy:*** Nevyžadují nepřetržitou přítomnost obsluhy na pracovišti. Může se jednat například o situaci, kdy je obsluha na stanovišti během pracovní doby podniku a po opuštění prostor zaměstnanci dojde i k odchodu obsluhy dohledového centra. Nicméně kamerový záznam se pořizuje dále a použil by se v případě, kdyby došlo k nějaké mimořádné události, byla nahlášena škoda, krádež apod. Může to být na úkor bezpečnosti provozu, ale daný systém snižuje náklady na lidské zdroje.
- ***Automatické systémy:*** Obejdou se bez nepřetržité přítomnosti obsluhy na stanovišti. Lidskou činnost zde suplují algoritmy umělé inteligence v softwaru, kterým jsou dohledová centra vybavena. Algoritmy vyhodnocují obraz v reálném čase a obsluhu informují jen v případě, kdy v záznamu vyhodnotí anomálii, která by mohla vést k ohrožení bezpečnosti provozu. Autonomní systém je dražší z hlediska vstupní investice a z důvodu využití většího množství kamer za účelem zabezpečení komplexního pokrytí střežené oblasti. (Deep Sentinel, 2023) (Šalomon, 2021)

4.2 Vývoj video dohledových systémů

Historie dohledových center sahá do Londýna 19. století, ale to ještě samozřejmě nebylo vybaveno kamerovou technikou. Dohledové centrum, tak jak ho známe dnes, bylo možno poprvé dohledat v nacistickém Německu. Již v roce 1942 ve velké míře začali používat kamerové systémy k zabezpečení a dokumentování procesu vývoje a testování raket V-2

v Peenemünde. Bohužel se, vzhledem k povaze provozu a množství dochovaných materiálů, nedochovalo mnoho informací.

Po 2. světové válce se ovšem kamery v zabezpečovacích systémech začaly objevovat v čím dál větší míře. Největší zastoupení měly zprvu v zabezpečení věznic, bank anebo přistávacích drah na letištích. Roli v tom hrála cena kamer a jejich rozměry. Kamera a zařízení pro nahrávání se ale postupem času staly sofistikovanější a umožňovaly reálný dohled a analýzu dat.

V posledních letech vývoj digitálních a síťových technologií vedl ke vzniku pokročilých dohledových center, která jsou schopna monitorovat a analyzovat velké množství dat z různých zdrojů, včetně kamer, senzorů a dalších zařízení. Během velkých akcí, jako jsou politické shromáždění, sportovní události a koncerty, přichází ke slovu algoritmy na vyhodnocování obrazu, které vyhodnocují chování účastníků, sbírají biometrické údaje a v případě anomálie či shody postupují dle předem naprogramovaných vzorců.

Obr. 5 Dohledové centrum společnosti M2C Space. (Zdroj: Největší komerční dohledové centrum střední Evropy je v Praze, 2023)

Dohledová centra hrají klíčovou roli ve veřejné bezpečnosti a jsou používána policií, armádami, bezpečnostními organizacemi a soukromými společnostmi k monitorování a reakci na potenciální hrozby. (Němeček, 2008)

4.3 Softwarové vybavení video dohledového systému

Výběr správného softwarového vybavení pro video dohledové centrum závisí na velikosti a komplexnosti systému, požadavcích na funkce a potřebách uživatelů. Je důležité vybrat

software od důvěryhodného výrobce a zajistit, aby byl software pravidelně aktualizován a zabezpečen.

Softwarové vybavení video dohledového centra by mělo obsahovat následující komponenty:

Software pro správu systému: Jedná se o software umožňující správci nebo uživateli řídit a spravovat video dohledový systém. Zástupci mohou být například Milestone XProtect, Dahua SmartPSS, Vivotek ST7501. Obvykle nabízejí následující funkce:

- ***Správa zařízení:*** Software pro správu systému by měl umožňovat správu kamer, síťových zařízení a dalšího hardwaru souvisejícího s video dohledem.
    - Nastavení přístupových práv: Umožňovat nastavit přístupová práva pro jednotlivé uživatele a definovat, kdo má přístup k jednotlivým funkcím a datům.
    - Monitorování stavu systému: Monitoruje stav systému, včetně stavu kamer a síťových zařízení, a upozorňovat správce na jakékoli potenciální problémy.
    - Archivace dat: Archivace videa a dalších dat získaných z video dohledového systému.
    - Zobrazení dat: Umožňuje zobrazení dat z video dohledového systému včetně zobrazení videa z jednotlivých kamer na jednom místě.
- ***Nástroje pro analýzu videa:*** umožňuje automaticky analyzovat a zpracovávat video získané z video dohledového systému. Tyto softwary obvykle nabízejí následující funkce:
    - Vyhledávání: Zajišťuje vyhledávání určitých objektů nebo akcí v archivech videa.
    - Detekce chování: Měl by umožňovat detekci předem naprogramovaného neobvyklého nebo narušujícího chování, jako je například krádež, vandalství nebo teroristický akt.
    - Analýza pohybu: Software by měl umožňovat sledovat a analyzovat pohyby objektů a lidí v rámci videa.
    - Automatické výstražné hlášení: Automatické generování výstražného hlášení v případě, že dojde k detekci neobvyklého nebo narušujícího chování.
    - Integrace s dalšími systémy: Důležitá vlastnost umožňující zefektivnit fungování celého centra díky kompatibilnosti jednotlivých programů.

Tyto funkce nabízí například produkty Video Insight, Genetec Security Center, Avigilon Control Center.

- ***Software pro archivaci videa:***
    - Ukládání videa: Spravovat ukládání videa na pevný disk, síťové úložiště nebo jiný typ úložiště.
    - Komprese videa: Program by měl umožňovat kompresi videa, aby se snížila jeho velikost a zlepšila efektivita úložiště.
    - Řízení přístupu: Zajišťuje, že pouze oprávněné osoby mají přístup k videu.
    - Vyhledávání videa: Umožňuje snadné vyhledávání videa podle různých kritérií, jako je čas, místo nebo událost.
    - Záloha videa: Zálohování videa, aby se zajistilo, že videa jsou bezpečně uložena a mohou být obnovena v případě nutnosti.

Z široké nabídky si lze vybrat produkty, jako jsou Nuuo Main Console, QNAP Surveillance Station, Synology Surveillance Manager.

- ***Software pro vzdálený přístup:*** Tento software by měl umožňovat vzdálený přístup k systému a zkombinovat funkci všech výše zmíněných produktů.
    - Připojení: Umožňuje uživatelům se připojit k video dohledovému systému z jakéhokoliv místa s připojením k internetu.
    - Přehrávání videa: Přehraje živé záběry nebo archivované video z video dohledového systému.
    - Správa systému: Uživatelé mohou spravovat systém z vzdáleného místa, například nastavit kameru, změnit nastavení systému nebo stahovat videa.
    - Bezpečnost: Primárně by měl poskytovat vysokou úroveň bezpečnosti, aby se zabránilo neoprávněnému přístupu k video dohledovému systému.

Mimo programů přímo dodávaných k některým kamerovým systémům lze zvolit například tyto iVMS-4200, RemoteView, Axxon Next.

- ***Software pro integraci s dalšími systémy:*** Jedná se o software umožňjicí integraci s dalšími systémy, jako jsou například systémy pro správu budov nebo bezpečnostní systémy, za účelem vytvoření komplexního systému, díky kterému bude dosaženo

maximální efektivity a nebudou kladeny příliš vysoké nároky na obsluhu. Poskytují tyto funkce:

- Integrace: Software umožňuje propojení video dohledového systému s jinými systémy, jako jsou například systémy pro zabezpečení budov nebo software dodávaný k jednotlivým komponentům dohledového systému.
- Automatické reakce: Může automaticky reagovat na události z jiných systémů, jako jsou například alarmy nebo detektory pohybu, a odeslat upozornění nebo zahájit živý přenos videa.
- Centralizované správa: Integrace s dalšími systémy umožňuje centralizovanou správu všech systémů prostřednictvím jednoho uživatelského rozhraní.

Orientačně lze zvolit z programů, jako jsou ExacqVision, Genetec Omnicast nebo Avigilon Integration Platform. Zásadní je pečlivě zvážit všechny možnosti a provést výběr software od důvěryhodného výrobce, který poskytuje podporu a aktualizace. (Software kamerových a dohledových systémů, 2023)

DÍLČÍ ZÁVĚR TEORETICKÉ ČÁSTI

Na základě rešerše zdrojů provedené za účelem sepsání předchozích kapitol je zřejmé,

že kybernetické zabezpečení subjektu musí být řešeno komplexně, aby výsledná opatření pokryla všechna rizika hrozící danému subjektu. Prvky kybernetického zabezpečení využívají nejnovější „know-how“ v oboru a neustále se vyvíjejí. Tento kontinuálně trvající vývoj je způsoben snahami o nalezení různých způsobů, jak dané zabezpečovací systémy pokud možno nepozorovaně překonat a dostat se tak k předmětům či informacím v nich ukrytých, neustávají. Záleží jen na tom, kdo má k dispozici jaké prostředky.

Další část této práce se soustředí na návrh algoritmů sloužící k zvýšení bezpečnosti video dohledového systému subjektu. Algoritmy budou zvoleny na základě studia dostupných produktů na trhu s programy zabývajícími se video dohledem.

II. PRAKTICKÁ ČÁST

5 PROGRAMY DOSTUPNÉ PRO VIDEO DOHLEDOVÉ SYSTÉMY

Jelikož se tato práce zaměřuje na využití AI v kontextu kybernetického zabezpečení se zacílením na využití prvků AI v zabezpečení VDS a vyhodnocování obrazu, je nutné vyhodnotit aktuálně nabízené programy, které těmito schopnostmi disponují.

5.1 Programy určené k vyhodnocení kamerového záznamu

Tyto programy lze v zásadě rozdělit na komerční (placené) a volně dostupné. Volně dostupné nejsou jen freeware programy, ale i tzv. open source software. Od toho se odvíjí možnosti a schopnosti programu. Největší výhodou open source programů je to, že za jejich vývojem stává komunita nadšenců, kteří se v dané oblasti pohybují, nezřídka pracují oficiálně i na placených verzích. Věnují vývoji svůj volný čas a sdílejí tím své „know-how“. Další důležitým přínosem je, že open-source programy jsou potenciálně transparentnější. Kódy těchto produktů nejsou chráněny obchodním tajemstvím, každý uživatel má tak možnost odhalit přítomnost škodlivého obsahu (např. neautorizované odesílání dat třetím stranám), jakožto i přispět tvorbou bezpečnostních záplat na odhalené nedostatky zdrojového kódu. Obojí je komunitou okolo open source licencovaných produktů aktivně využíváno.

5.1.1 Komerční programy

Existuje mnoho komerčních programů k vyhodnocování obrazu, které jsou využívány v bezpečnostních kamerových systémech. Všechny tyto programy spojuje to, že umožňují detekci pohybu a sledování objektů. Některé z nejznámějších programů jsou:

- ***Milestone XProtect*** – Tento produkt nabízí pokročilé funkce, jako je rozpoznávání registračních značek, detekce obličejů a analýza chování. Má také možnost přidání velkého množství doplňků (tzv. add-on), kterými si každý uživatel může program specifikovat dle svých potřeb. (Software kamerových a dohledových systémů, 2023)
- ***Genetec Security Center*** – Tento program umožňuje integraci s mnoha různými typy kamerových systémů a také nabízí možná rozšíření o pokročilé funkce, jako je detekce zbraní a analýza chování. (Genetec Inc., 2023)
- ***Avigilon Control Center*** – Nabízená řada programů od firmy Avigilon využívá pokročilých algoritmů ke zpracování obrazu a nabízí nejen řadu funkcí, ale také ucelené modelové řady kamer a čidel určených k zabezpečení objektů. Dále

spolupracuje se společností Motorola a podílí se na vývoji softwaru pro jejich produkty. (Avigilon, 2023)

- ***Bosch Video Management System*** – Stejně jako výše zmíněné produkty, i tento nabízí řadu funkcí pro sledování kamerových systémů. Společnost Bosch navíc aktivně pracuje s ONVIF knihovnami a lze tak pomocí její aplikace ovládat řadu kamer jiných výrobců. (Bosch Video Management Software, 2023)
- ***Hikvision iVMS-4200*** – Také tento program umožňuje integraci s mnoha různými typy kamerových systémů a nabízí širokou škálu funkcí. Z tohoto seznamu má nejblíže k běžnému koncovému uživateli. (IVMS - 4200 Software - Hikvision, 2023)

Je však důležité si uvědomit, že výběr správného programu k vyhodnocování obrazu závisí na konkrétních potřebách uživatele a vlastnostech konkrétního kamerového systému, který se bude používat. Také se to odvíjí od investovaných prostředků. Nějaký základní software dnes ke svým produktům nabízí v podstatě každý výrobce kamer, záleží tak doopravdy jen na požadavcích zákazníka a jeho rozpočtu.

5.1.2 Open source software

Existuje také řada open source programů k vyhodnocování obrazu, které jsou k dispozici zdarma a mohou být použity pro vytvoření vlastního bezpečnostního kamerového systému. Svými schopnostmi a nabízenými funkcemi nijak nezaostávají za komerčními produkty. Jen je pro spoustu uživatelů kamerových systémů jednodušší si nainstalovat software dodaný s produktem. A také zde hraje roli bezpečnost. Zvlášť u firem raději zaplatí za produkt, u kterého mají záruku původu a zákaznického servisu. Některé z nejznámějších open source programů jsou:

- ***ZoneMinder*** – Jedná se o komplexní platformu pro správu kamerových systémů, která umožňuje snadnou integraci s mnoha různými typy kamer. Je schopna zajistit fungování jakékoliv dnes nabízené kamery s širokou paletou možných doplňků. Má velice velkou fanouškovskou, a tudíž i vývojářskou základnu. Nicméně je zajímavé, že ačkoliv je aplikace zdarma, tak servisní podpora je placená. (ZoneMinder, 2023)
- ***iSpy*** – Tento video dohledový software umožňuje sledovat a ovládat kamery na nejznámějších platformách jako je Windows, iOS a Linux. Je open source, ale pro komerční využití je určená placená verze. (ISpy, 2023)

- ***Shinobi*** – Program je navržen tak, aby byl snadno použitelný pro vytvoření vlastního bezpečnostního kamerového systému a stejně jako iSpy je k dispozici i jeho placená verze. (Shinobi, 2023)
- ***MotionEye*** – Tento program poslední dobou nabývá na popularitě díky svému webovému rozhraní a hardwarové nenáročnosti. (GitHub, 2023)

Většina moderních open source programů k vyhodnocování obrazu umožňuje implementaci prvků umělé inteligence. Například programy jako *OpenCV, TensorFlow* nebo *PyTorch* umožňují snadné vytvoření aplikací s detekcí objektů, rozpoznáváním obrazů a jinými funkcemi zpracování obrazu, které využívají umělou inteligenci.

Mezi konkrétní projekty v rámci open source programů, které využívají umělou inteligenci, patří například:

- ***YOLO (You Only Look Once)*** – populární algoritmus pro detekci objektů, který umožňuje rychlé a přesné určení polohy a typu objektů na obrazu. (YOLO, 2023)
- ***Mask R-CNN*** – Algoritmus, který umožňuje detekovat a segmentovat objekty na obrazu a přiřadit jim nějakou konkrétní kategorii. (GitHub, 2023)
- ***DeepStack*** – Open source nástroj pro rozpoznávání obrazů a detekci objektů s využitím neuronových sítí a strojového učení. (Moravčík et al., 2017)

Tyto projekty a mnoho dalších umožňují integraci umělé inteligence do bezpečnostního kamerového systému a výrazně zlepšit jeho funkce.

5.2 Možnosti úpravy programu pomocí programovacího jazyka Python

V další části této práce bude využit programovací jazyk *Python*, knihovny *OpenCV* a *Tkinter*, a proto zde jsou ve stručnosti popsány.

5.2.1 Programovací jazyk Python

V současnosti se jedná o jeden z nejpoužívanějších programovacích jazyků na světě. Je to vysokoúrovňový, interpretovaný, objektově orientovaný jazyk, který je vhodný pro různé účely od webových aplikací až po vědecké výpočty a umělou inteligenci. Jeho popularita spočívá v jednoduchosti používání, přívětivé syntaxi a široké nabídce knihoven a modulů, které usnadňují práci programátora. Jednou z největších výhod Pythonu je jeho přehlednost a čitelnost kódu. Syntax Pythonu je velmi jednoduchá a čitelná, což znamená, že se noví

uživatelé mohou rychle naučit programovat v tomto jazyce. Python je také známý pro své dynamické typování. Proměnné nemusí být definovány předem a mohou být přiřazovány různé typy dat v průběhu běhu programu.

Jedním z nejvýznamnějších použití Pythonu je webový vývoj. Python nabízí mnoho frameworků pro webové aplikace, jako je například *Django* a *Flask*. Tyto frameworky usnadňují vývoj webových aplikací a zjednodušují mnoho z běžných úkolů, jako jsou například práce s databázemi, autentizace uživatelů a vytváření rozhraní API.

Python také nabízí širokou škálu knihoven a modulů pro práci s daty, včetně knihoven jako *Pandas*, *NumPy* a *Matplotlib*. Tyto knihovny umožňují programátorům snadno pracovat s daty a provádět různé analýzy, výpočty a vizualizace. Díky tomu se Python stal velmi oblíbeným nástrojem pro data science a strojové učení.

Python se také používá pro automatizaci různých úloh, například pro tvorbu skriptů pro správu souborů, síťového provozu a mnoho dalšího. Python také poskytuje rozsáhlou knihovnu pro práci s textem, což umožňuje programátorům vytvářet skripty pro zpracování a analýzu textových dat.

Neposlední výhodou Pythonu je jeho rozšiřitelnost. Python podporuje volání funkcí napsaných v jiných jazycích, jako jsou C a C++, což umožňuje programátorům využívat stávající knihovny napsané v těchto jazycích.

Volání může probíhat formou:

- ***Volání funkce přeložené jako Python modul:*** Tento přístup spočívá v přeložení kódu napsaného v jiném jazyce jako Python modulu. Modul je poté načten do Pythonu a funkce jsou volány jako běžné funkce v Pythonu.
- ***Volání funkce pomocí rozhraní API:*** Některé jazyky, jako je například C, nabízejí rozhraní pro volání funkcí z jiných jazyků. Tyto funkce mohou být volány pomocí speciálních funkcí Pythonu, které jsou schopny komunikovat s rozhraním API a volat funkce.
- ***Volání funkce pomocí externího procesu:*** V některých případech může být nutné spustit funkci napsanou v jiném jazyce jako externí proces. Tento proces je spuštěn z Pythonu a komunikuje s ním pomocí standardního vstupu a výstupu.

Python také nabízí možnosti pro vytváření GUI aplikací. K tomuto účelu mohou programátoři použít knihovny jako *Tkinter*, *PyQt* nebo *wxPython*, které umožňují vytváření grafického uživatelského rozhraní.

Další výhodou Pythonu je jeho obrovská komunita. Existuje mnoho uživatelů Pythonu, kteří přispívají k vývoji knihoven a modulů a poskytují podporu pro začínající programátory. Existují také různé online zdroje, jako jsou fóra, tutoriály a videa, které pomáhají novým uživatelům rychle se naučit programovat v Pythonu.

V poslední době se Python také stává stále populárnějším jazykem pro vývoj aplikací pro umělou inteligenci a strojové učení. K tomuto účelu jsou k dispozici knihovny jako *Tensorflow*, *PyTorch* nebo *Scikit-learn*, které umožňují vytvářet a trénovat modely pro různé úlohy jako jsou například klasifikace, detekce objektů, překlad jazyků a mnoho dalších. (PythonBooks - Python Wiki, 2023)

5.2.2 OpenCV

OpenCV (Open Source Computer Vision Library) je open-source knihovna pro počítačové vidění, zpracování obrazů a strojové učení. Je navržena pro využití v různých aplikacích, jako jsou například robotika, detekce objektů, zpracování obrazů v reálném čase a mnoho dalších.

OpenCV je napsána v jazyce *C++*, ale nabízí rozhraní pro mnoho jazyků, včetně Pythonu. Díky tomu může být *OpenCV* použita v mnoha různých prostředích a aplikacích.

Funkcionality, které *OpenCV* nabízí, zahrnují například:

- ***Zpracování obrazu:*** *OpenCV* poskytuje nástroje pro zpracování obrazu, jako je například filtr pro snížení šumu v obraze, transformace obrazu a mnoho dalších.
- ***Detekce objektů:*** *OpenCV* obsahuje algoritmy pro detekci objektů v obraze, jako jsou například detekce obličejů, detekce pohybu a mnoho dalších.
- ***Strojové učení:*** *OpenCV* obsahuje nástroje pro strojové učení, jako jsou například klasifikační algoritmy, neuronové sítě a mnoho dalších.
- ***Zpracování videa:*** *OpenCV* obsahuje nástroje pro zpracování videa v reálném čase, jako jsou například detekce objektů v pohybu, sledování objektů a mnoho dalších. (Home - OpenCV, 2023)

5.2.3 TensorFlow

TensorFlow je knihovna pro strojové učení a hluboké učení, která byla vytvořena společností *Google*. Umožňuje vývojářům vytvářet a trénovat neuronové sítě a modely strojového učení.

Obsahuje nástroje pro tvorbu grafů výpočtů, kde každý uzel představuje operaci s daty. Grafy výpočtů lze následně spustit na různých výpočetních jednotkách, jako jsou CPU, GPU nebo TPU.

Lze v něm trénovat modely pomocí nejnovějších algoritmů hlubokého učení, jako jsou konvoluční neuronové sítě, rekurentní neuronové sítě, autoenkodéry a další. Knihovna také obsahuje funkce pro práci s daty, jako je načítání dat, předzpracování dat a vytváření trénovacích a validačních datových sad. Tyto modely zde lze vytvářet a trénovat s různými úrovněmi složitosti. Od malých modelů pro klasifikaci obrazů a textu až po velké modely pro rozpoznávání řeči a překlad jazyků. Díky svému modulárnímu návrhu a výkonným výpočetním schopnostem se stal populární knihovnou pro vývojáře strojového učení a hlubokého učení po celém světě. Dalším kladem je podpora široké komunity vývojářů a z toho plyne existence množství materiálů, kurzů a knih, které umožňují rychle a snadno získat základní znalosti o práci s touto knihovnou. (TensorFlow, 2015)

5.2.4 Tkinter

Tkinter je knihovna pro tvorbu grafického uživatelského rozhraní (GUI) v jazyce Python. Je to součást standardní knihovny Pythonu, takže není nutné ji instalovat samostatně.

Tkinter používá knihovnu *Tk*, což je multi-platformní toolkit pro tvorbu GUI napsaný v jazyce *Tcl* (Tool Command Language). *Tkinter* poskytuje pythonovské rozhraní pro používání této knihovny, takže programátoři mohou vytvářet GUI aplikace v Pythonu pomocí nástrojů a widgetů poskytovaných *Tk*.

Tkinter obsahuje mnoho základních widgetů, jako jsou tlačítka, vstupní pole, rolovací seznamy, nabídky a mnoho dalších. Widgety mohou být snadno umístěny na hlavním okně aplikace a interagovat s uživatelem pomocí událostí a funkcí callback.

Tkinter také umožňuje vytvářet vlastní widgety a rozšiřovat stávající widgety pomocí metod a funkcí, které jsou poskytovány knihovnou *Tk*.

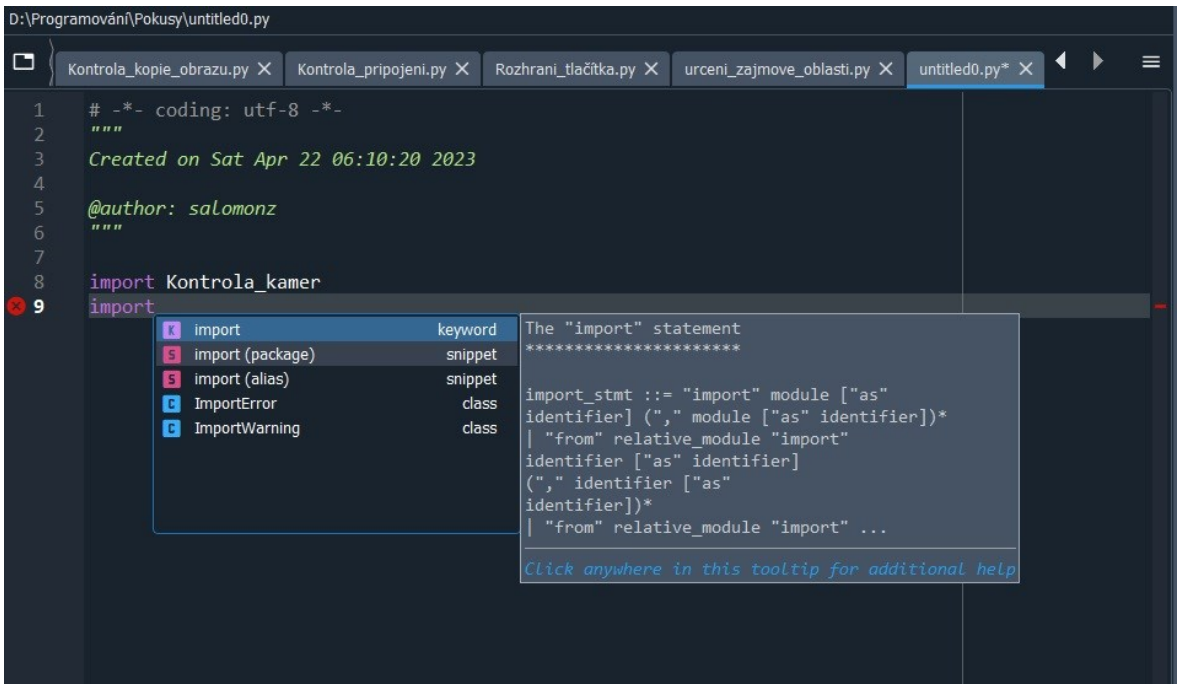
Tkinter je často používán pro tvorbu desktopových aplikací, jako jsou textové editory, grafické nástroje, hry a mnoho dalších. *Tkinter* je snadno ovladatelná a je vhodná pro rychlou tvorbu prototypů aplikací, což ji činí populární volbou pro začátečníky i zkušené vývojáře.

Pomocí *Tkinter* lze také vytvářet aplikace, které využívají další knihovny Pythonu, jako jsou například *NumPy* pro zpracování dat nebo *OpenCV* pro zpracování obrazu. *Tkinter* tedy umožňuje integraci různých nástrojů a knihoven do jediné aplikace s grafickým uživatelským rozhraním. (Tkinter - Python interface to Tcl/Tk, 2023)

5.2.5 Spyder

Algoritmy byly napsány a testovány v open source vývojovém prostředí (IDE) Spyder pro jazyk Python, které bylo navrženo s cílem usnadnit tvorbu, testování a ladění kódu. Jeho uživatelské rozhraní je intuitivní a nabízí mnoho funkcí, jako jsou editory kódu, konzole, debugger a správce projektů.

Spyder má mnoho vlastností, které z něj činí populární IDE pro vývojáře. Například editor kódu nabízí rychlé automatické dokončování kódu a kontrolu syntaxe, takže programátoři mohou rychle psát kód bez nutnosti manuálního vkládání kódu a zamezit tak mnoha běžným chybám. Konzole umožňuje programátorům spouštět kód, zkoumat výstupy a interaktivně testovat funkce. Nabízí také vestavěný debugger, což zjednodušuje hledání a opravu chyb v kódu.



Obr. 6 Ukázka prostředí IDE Spyder (Zdroj: Vlastní, 2023)

Spyder má také funkci pro správu projektů, která umožňuje uživatelům snadno organizovat kód a soubory v rámci projektu. To usnadňuje práci na velkých projektech, které mohou zahrnovat tisíce řádků kódu.

Bonusem je také to, že obsahuje integrovanou dokumentaci, která umožňuje uživatelům snadno přistupovat k dokumentaci Python knihoven a funkcí, což může být velmi užitečné pro nové programátory, kteří se snaží zjistit, jak použít určitou funkci.

Celkově lze říci, že Spyder je vynikající vývojové prostředí pro Python vývojáře, kteří hledají intuitivní a funkční prostředí pro tvorbu, testování a ladění kódu. Je snadno použitelný, přestože nabízí mnoho pokročilých funkcí a vlastností, které usnadňují práci na projektech jak pro začátečníky, tak i pro zkušené programátory. (IDE Spyder, 2023)

6 NÁVRH ALGORITMŮ URČENÝCH K IMPLEMENTACI DO VIDEO DOHLEDOVÉHO SYSTÉMŮ

Tato část práce se bude zabývat tvorbou algoritmů, které bude možno díky API použitých kamer aplikovat během pořizování záznamu a zvýšit tak bezpečnost a využitelnost dohledového systému.

6.1 Návrh algoritmů

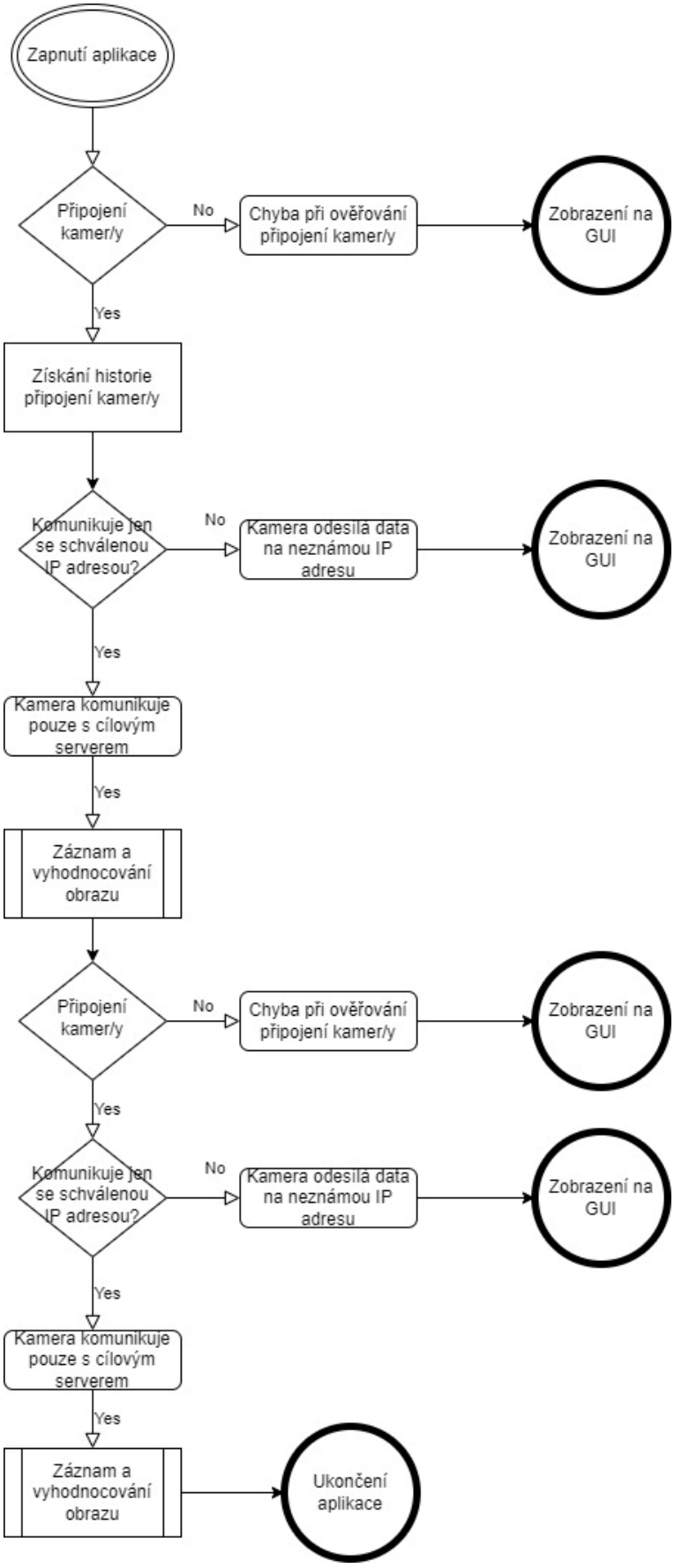
Při navrhování algoritmů hraje významnou roli komplexnost problému a množství vstupů. Proto je důležité pokusit se tyto kódy nejprve matematicky či logicky namodelovat a udržet tak řešený problém v mezích realizovatelnosti. Navyše tento proces vyžaduje kreativitu a experimentování, protože neexistuje žádná univerzální metoda pro řešení všech problémů. Jak již bylo zmíněno, dalším důležitým faktorem, který může ovlivnit návrh algoritmů, je množství vstupů, na které algoritmus musí reagovat. Čím více vstupů a situací musí algoritmus zvládnout, tím složitější může být jeho návrh a implementace. Vstupní data mohou být různorodá a náročná na zpracování, což může vést k více iteracím a testování algoritmu. To se následně promítne do časové náročnosti testování a ladění algoritmů.

6.1.1 Kontrola přenosu dat z kamery

První algoritmus se bude zabývat kontrolou, zda je připojení kamer/y stále zabezpečené a nemá k ní přístup nikdo jiný. Nebude ovšem suplovat práci anti-virového programu nebo firewalu, ale bude ověřovat, zda je signál z kamery odesílán jen na určené IP adresy a data z ní neputují nikam jinam. Návrh začne vytvořením vývojového algoritmu, kde bude specifikováno, co bude po algoritmu požadováno a také poslouží ke kontrole během tvorby.

1. Kód začíná importem potřebných knihoven, což jsou *tensorflow*, *numpy*, *cv2*, *time* a **socket*. Tyto knihovny umožňují práci s neuronovými sítěmi, numerickými výpočty, zpracování obrazu, časem a síťovým spojením.
2. V dalších řádcích kódu se definují proměnné pro načtení sítě a vážených hodnot z předem natrénovaného modelu, který se používá pro detekci objektů. Tento model je součástí knihovny *tensorflow* a umožňuje detekovat objekty v reálném čase.

3. Následuje inicializace proměnných pro práci s kamerou a vstupním signálem. Zde se také definuje port, na kterém se bude ověřovat spojení kamery. Při úspěšném ověření kamery se na ní bude sledovat pohyb objektů.
4. Další část kódu se zabývá funkcemi pro ověřování spojení kamery. Funkce *check_camera_connection()* slouží k otestování, zda kamera správně odpovídá na dotaz přes socketové spojení. Pokud kamera odpovídá, vrátí se hodnota True, jinak False.
5. Funkce *check_camera()* pak volá předchozí funkci a vypisuje výsledek ověření. Pokud se podaří ověřit spojení kamery, vypíše se „Kamera připojena“, v opačném případě „Chyba při ověřování připojení kamery“.
6. Hlavní část kódu se zabývá sledováním objektů. V první řadě se otevře vstupní signál z kamery, který se následně zpracovává. Pro zpracování signálu se používá cyklus, který běží neustále a zpracovává nové snímky.
7. V každém průchodu cyklem se nejprve ověří spojení kamery voláním funkce *check_camera()*. Pokud ověření selže, cyklus se zastaví. V opačném případě se pokračuje s detekcí objektů.
8. Detekce objektů se provádí pomocí předem natrénovaného modelu. Zpracování snímků se provádí v cyklu pomocí funkce *detect_objects()*, která vrací seznam objektů, které byly nalezeny na snímku.
9. Vstupní signál obvykle představuje obraz nebo video stream z kamery, který je dále zpracováván algoritmy počítačového vidění. V tomto kódu se na začátku ověřuje, zda je vstupní signál přítomen, tedy zda kamera posílá data. Pokud ne, znamená to, že kamera není připojena, nebo je připojena špatně a není možné pokračovat v zpracování obrazu. Pokud je signál přítomen, znamená to, že kamera funguje správně a je možné pokračovat v programu.



Obr. 7 Návrh algoritmu kontrolující komunikaci kamery (Zdroj: Vlastní, 2023)

6.1.2 Kontrola připojení kamery

Tento algoritmus slouží pouze ke kontrole správné funkce přenosu dat z kamery. Lze v něm nastavit interval kontroly a stejně jako v předchozích algoritmech je možné, vytvořit si seznam kamer, které budou kontrolovány.

1. Nejprve se v kódu importuje knihovna *cv2*. Tato knihovna je knihovnou pro práci s obrazem a videem.
2. Dále se v kódu nachází seznam kamer, které jsou připojeny k systému. Tyto kamery jsou specifikovány pomocí URL adres. V této části kódu se využívá seznam, aby se zjistilo, zda je každá kamera připojena a funguje.
3. V cyklu *for* jsou všechny kamery v seznamu zkontrolovány, zda jsou připojeny a připraveny k použití. Pro každou kameru se používá funkce *cv2.VideoCapture()* k otevření streamu obrazu. Pokud se podaří spojení se streamem, zobrazí se zpráva, že kamera je připojena – „Kamera {číslo kamery} je připojena“. Pokud se kamera nepodaří připojit, zobrazí se zpráva, že kamera není připojena – „Kamera {cam} není připojena“ - a přeskočí se na další kameru v seznamu.
4. Další část kódu obsahuje funkci *check_camera_connection()*, která slouží k ověření, zda je kamera připojena a posílá obrazový stream. Funkce nejprve pomocí *cv2.VideoCapture()* otevře stream k dané kameře. Pokud se stream podaří otevřít, funkce načte jeden snímek pomocí *cap.read()*. Pokud se podaří načíst snímek, funkce vypíše zprávu, že kamera je připojena a posílá stream obrazu. Pokud se snímek nepodaří načíst, funkce vypíše zprávu, že kamera je připojena, ale není možné získat stream obrazu. Pokud se nepodaří otevřít stream kamerového zařízení, vypíše se zpráva, že se nepodařilo připojit kameru a předpokládá se, že není správně připojena nebo konfigurována.
5. Na konci kódu se funkce *check_camera_connection()* zavolá a provede se kontrola připojení kamery.

6.1.3 Tvorba rozhraní pro určení kontrolní oblasti

Pro zaměření určité části obrazovky je třeba znát souřadnice této oblasti v rámci obrazu. Ty mohou být buď pevně nastaveny v kódu, nebo mohou být získány interaktivně od uživatele (např. kliknutím myší na rohy oblasti).

1. Kód opět využívá knihovnu OpenCV, v první řadě je načtena knihovna pomocí příkazu *import cv2*. Poté následuje definice funkce *select_roi*, která slouží k výběru oblasti zájmu kliknutím myši.
2. Tato funkce má jako argumenty *event, x, y, flags* a *params*. Tyto argumenty jsou předávány funkci automaticky z OpenCV. Funkce *select_roi* pracuje s globálními proměnnými *roi, roi_selected, start_x, start_y, end_x* a *end_y.* Tyto proměnné jsou použity v hlavní smyčce programu pro určení oblasti zájmu.
3. Po definici funkce *select_roi* následuje nastavení kamery pomocí příkazu *cap = cv2.VideoCapture(0)*. Tím se inicializuje objekt pro čtení z kamery. Pokud má uživatel více kamer připojených, může změnit číslo parametru z 0 na index kamery, kterou chce použít.
4. Poté je nastavena callback funkce pro výběr ROI pomocí *cv2.namedWindow("frame")* a *cv2.setMouseCallback("frame", select_roi)*. Funkce *cv2.namedWindow* vytváří okno s názvem "frame", ve kterém se zobrazí snímek z kamery. Funkce *cv2.setMouseCallback* nastavuje funkci *select_roi* jako callback pro události myši na snímku.
5. Nyní kód vstupuje do hlavní smyčky programu pomocí while True:. Tato smyčka bude cyklicky načítat snímky z kamery a zobrazovat je v okně "frame". Nejprve se pomocí *cap.read()* načte snímek z kamery do proměnné frame. Poté se testuje, zda byl snímek úspěšně načten pomocí if ret:.
6. Pokud byla vybrána oblast zájmu pomocí funkce `*select_roi*, snímek se nejprve ořízne na velikost této oblasti a poté se aplikuje funkce *detect_motion*, která zpracuje snímek a vrací oblast s největším pohybem. Funkce *detect_motion* pracuje s dvěma snímky: aktuálním snímkem z kamery a předchozím snímkem, který slouží k určení změny mezi snímky. Funkce nejprve konvertuje obraz na stupně šedi, spočítá rozdíl mezi snímky, binarizuje rozdíl a vyhledá největší konturu v binarizovaném obrazu. Na základě největší kontury se určí oblast zájmu a vrátí se jako výstup funkce. Tato oblast se následně vyznačí na původním snímku pomocí funkce *cv2.rectangle*.
7. Kód dále obsahuje nekonečnou smyčku while True, která se opakuje, dokud není program ukončen. V této smyčce se neustále načítají snímky z kamery pomocí funkce *cap.read()*, které jsou zpracovávány stejným způsobem jako popsaný výše.

8. Program lze ukončit stisknutím klávesy "q", což způsobí přerušení smyčky a uvolnění kamery a okna.

Kód bude načítat video a pro každý snímek získávat část obrazu odpovídající definované oblasti ROI. Následně bude tato část zpracována a vykresleno ohraničení této oblasti na původním snímku pomocí funkce *cv2.rectangle*. Výsledný snímek je zobrazován v okně a program běží, dokud není stisknuta klávesa "q" nebo dokud nedojde k poslednímu snímku videa. Po skončení se uvolní zdroje pomocí funkcí *video.release()* a *cv2.destroyAllWindows()*.

6.1.4 Kontrola smyčky

Samotný kód na detekci smyčky v kamerovém záznamu není příliš rozsáhlý, ale kromě něj jsou zde také další části kódu na grafické rozhraní, interakci s uživatelem a integraci s kamerou. Zde je relativně podrobný popis kódu určeného k detekci nahrané smyčky v obraze.

1. Funkce začíná tím, že otevře video soubor pomocí *cv2.VideoCapture*. Tato třída je součástí knihovny OpenCV a umožňuje přehrávání videa a čtení jednotlivých snímků.
2. Příkaz *get* s argumentem *cv2.CAP_PROP_FRAME_COUNT* slouží k získání celkového počtu snímků ve videu. Tato hodnota bude následně použita k rozdělení videa na menší segmenty.
3. Po získání počtu snímků se určí velikost jednoho segmentu. V této implementaci se využívá velikost 300 snímků, tedy zhruba 10 sekund videa při standardní rychlosti 30 snímků za sekundu. Tento segment pak slouží k porovnání s předchozím segmentem.
4. Po určení velikosti segmentu se vypočte počet segmentů v celém videu. Toto se provádí pomocí celočíselného dělení počtu snímků ve videu a velikosti segmentu. Výsledek tohoto dělení pak udává, kolik segmentů bude vytvořeno.
5. Vnitřní smyčka pak postupně prochází všechny snímky v jednotlivých segmentech a porovnává je s předchozím segmentem. Při porovnání se využívá funkce *cv2.absdiff*, která odečte jeden snímek od druhého a vypočítá absolutní hodnotu rozdílu.

   Tím se získává maska, která ukazuje, které pixely jsou rozdílné v obou snímcích.

Následně se vypočítá průměrná hodnota rozdílu mezi pixely, a pokud je rovna nule, znamená to, že jsou oba snímky totožné.

6. V případě nalezení kopie, se zobrazí oznamovací okno s pomocí třídy *tk.messagebox.showinfo* z knihovny *tkinter*. Tato třída slouží k zobrazení dialogového okna s daným textem. V tomto případě se jedná o okno s textem "Byla nalezena kopie v kamerovém záznamu!".
7. Po zobrazení oznamovacího okna se funkce ukončí a uvolní se zdroje. To se provádí pomocí metody release na objektu *cv2.VideoCapture*. Tato metoda uvolňuje všechny zdroje, které byly alokovány pro přehrávání videa.
8. Celkově tedy tato funkce umožňuje detekovat kopie v kamerovém záznamu a poskytuje jednoduché řešení pro identifikaci těchto kopii pomocí vizuálního upozornění. Když funkce narazí na stejný snímek v průběhu dvou různých segmentů videa, zobrazí se informační okno, které upozorňuje uživatele na detekovanou kopii. Toto okno obsahuje zprávu „Byla nalezena kopie v kamerovém záznamu!“ a uživatel ho může jednoduše zavřít.

Obr. 8 Návrh algoritmu k detekci smyčky v záznamu (Zdroj: Vlastní, 2023)

6.2 Návrh uživatelského rozhraní

Tato kapitola bude zaměřena na návrh algoritmů, které umožní zjednodušit ovládání vytvořených algoritmů při jejich testování a následném užívání. Pro návrh uživatelského rozhraní bude využita opět knihovna *Tkinter*.

6.2.1 Tvorba grafického rozhraní ovládání

Tvorba uživatelského rozhraní bude sestávat z těchto kroků:

1. Nejprve se musí importovat modul *Tkinter*.
2. Poté je vytvořeno hlavní okno aplikace.
3. V dalším kroku jsou vytvořena tlačítka potřebná k ovládání algoritmů.
4. Následně výběrový prvek pro volbu kamer, režim detekce a volbu čtverce.
5. Další je prvek pro hlášení textových zpráv.

Tento kód je implementací jednoduchého uživatelského rozhraní pomocí knihovny *Tkinter*, která umožňuje tvorbu grafických uživatelských rozhraní pro desktopové aplikace. V kódu je vytvořeno pět tlačítek, každé s vlastní funkcionalitou a textové pole.

- Funkce pro každé tlačítko jsou definovány jako samostatné funkce, které jsou poté přiřazeny k odpovídajícímu tlačítku pomocí atributu *command.* Když uživatel klikne na tlačítko, spustí se příslušná funkce.
- Každá funkce je implementována jako volání určitého algoritmu, který provádí kontrolu nějakého zařízení nebo procesu a vrací výsledek této kontroly. Výsledek je poté zobrazen v textovém poli pomocí metody *insert()*.
- Kromě tlačítek a textového pole obsahuje kód i vlastní funkci *run_algorithm()*, která je volána z každého tlačítka a která zajišťuje spuštění správného algoritmu a zobrazení výsledku.

6.2.2 Sloučení algoritmů

Pro sloučení našich algoritmů, do jednoho balíčku, můžeme použít modulární strukturu kódu. Využijeme toho, že každý algoritmus je vlastním samostatným modulem, který můžeme vložit do hlavního skriptu.

Algoritmy tedy pojmenujeme podle prováděných akcí: *GUI_kamery.py, pripojeni_kamer.py, kontrola_kopie_obrazu.py, kontrola_pripojeni.py, kopie_obrazu.py, rozhraní_tlačítka.py* a *urceni_zajmove_oblasti.py.*

Nyní nastaly dvě možnosti. První možností by bylo vytvoření hlavního skriptu s názvem *main.py*, do kterého budou vloženy všechny tyto moduly pomocí příkazu import. V hlavním skriptu bude fungovat vytvořené uživatelské rozhraní pomocí knihovny *Tkinter*, jak již bylo zmíněno v předchozí kapitole.

Tato struktura by umožnila snadné rozšiřování a úpravu jednotlivých modulů, aniž by se musel měnit kód v hlavním skriptu. To by usnadnilo údržbu kódu a zvyšovalo jeho přehlednost.

Druhá možnost, která byla nakonec vzhledem k zjednodušení obsluhy výsledného balíčku zvolena, je pomocí souboru *__init__.py* vloženého do složky s ostatními algoritmy. Tím bude vytvořen balíček, který může být snadno importován do jiných Python skriptů.

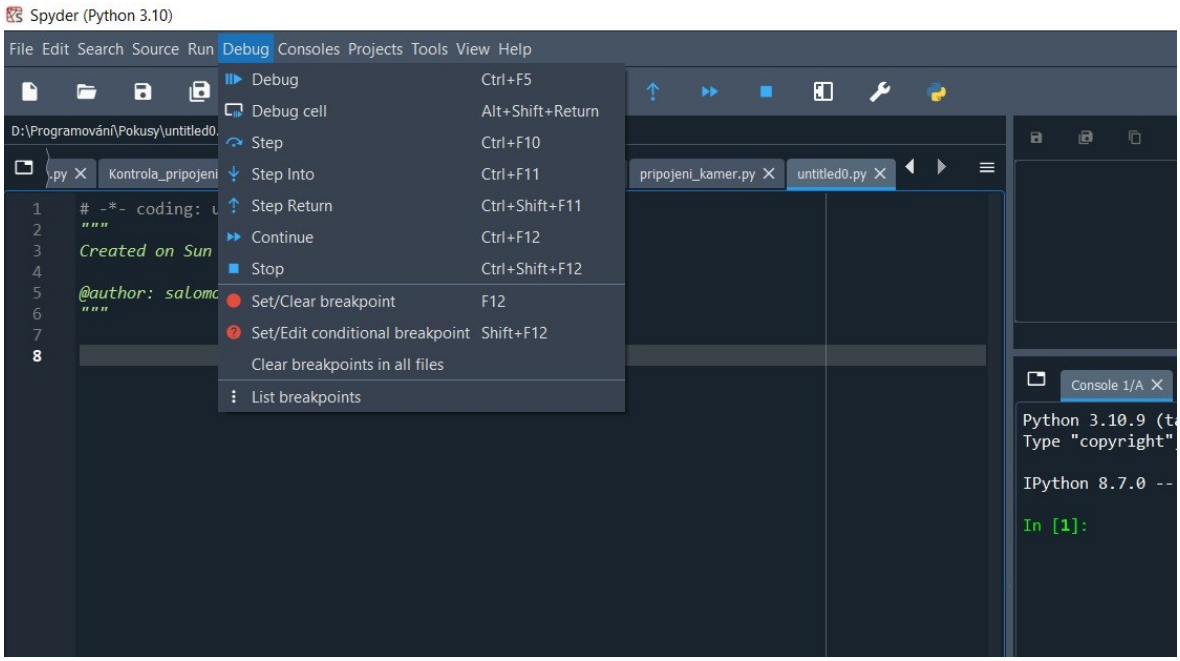
Jeho součástí bude také soubor *README.txt*, který bude obsahovat stručný popis kódů, jaké funkcionality nabízí a jakým způsobem jej lze použít. Tento soubor bude umístěn v téže složce jako kód.

Dále bude ve stejném adresáři zahrnut skript pro instalaci potřebných knihoven. Vzhledem k tomu, že kódy využívají externí knihovny, bude nutno vytvořit soubor *requirements.txt*, který obsahuje seznam všech knihoven potřebných pro spuštění kódu. Pomocí nástroje *pip* a příkazu *`pip install –r requirements.txt`* lze pak tyto knihovny jednoduše nainstalovat. Po souboru s požadavky bude následovat soubor *install.bat*, který bude obsahovat jednoduchý skript pro instalaci knihoven a spuštění kódu.

S takto uspořádanou strukturou souborů bude mít uživatel, který chce použít kód, jasný přehled o tom, jak kód nainstalovat a jak ho používat.

6.2.3 Ověření návrhů algoritmů

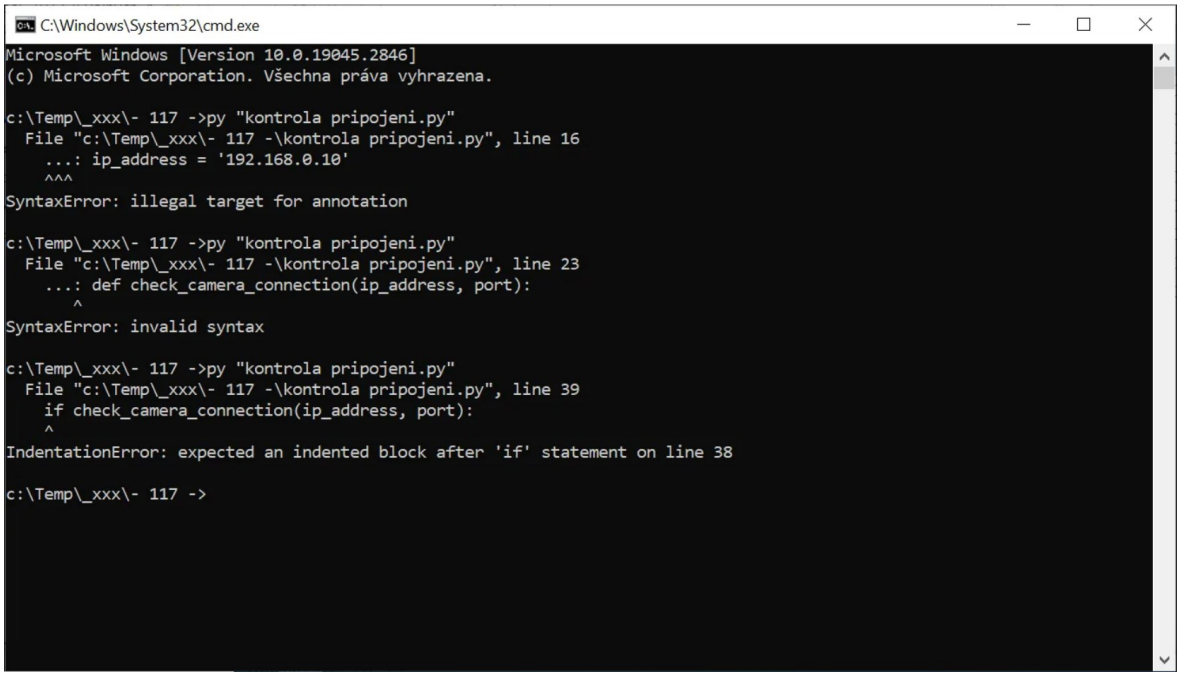
Zkušební provoz navržených kódů je časově náročná část, sloužící k ověření konceptu a správnosti návrhu. Část problémů spojených s tvorbou kódů v jazyce Python umožnilo eliminovat použití vývojového prostředí IDE Spyder, díky tomu, že v sobě má debugger a průběžně kontroluje napsaný kód a syntaxe v něm.



Obr. 9 Kontrola chyb v kódu v prostředí Spyder (Zdroj: Vlastní, 2023)

Debugger umožňuje několik zásadních akcí. První je nastavení bod zastavení v kódu, aby se kód zastavil v určitém místě a mohl se prozkoumat stav proměnných, tzv. *Breakpoints*. Dále také umožňuje postupně spouštět kód po jednotlivých řádcích, takže lze sledovat, jak se kód vyvíjí, tzv. *Stepping* a v neposlední řadě je užitečnou schopností to, že umožňuje sledovat hodnotu určité proměnné během běhu programu, tzv. *Watch*. (Spyder IDE, 2023)

Další výhodou zvoleného prostředí je jeho konzole, kde je možné vyzkoušet funkčnost vytvořeného kódu a ověřit si, zda výstupy odpovídají těm očekávaným. V případě, že se objeví chyba nesouvisející s kódem, ale jež vznikla v důsledku provádění kódu, je to na konzoli ve vývojovém prostředí vypsáno a lze tak přijmout potřebná opatření. Na obrázcích Obr. 10 a Obr. 11 je vidět významný rozdíl v prostředích. Obě zobrazují stejnou chybu, ale v případě IDE Spyder je vypsáno i to, co chybně zadaná IP adresa způsobí a jaké další kroky neproběhnou.



Obr. 10 Zobrazení chyb v běžném příkazovém řádku (Zdroj: Vlastní, 2023)

Obr. 11 Ukázka chybového hlášení v prostředí Spyder při zadání špatné IP adresy kamer (Zdroj: Vlastní, 2023)

Po odstranění chyb v kódu a na vstupech bylo možno přejít k testování funkce samotných algoritmů. Tato část je výsledkově značně různorodá, neboť u některých kódů je výsledek zřejmý hned zatímco, zejména v případě algoritmu hledajícího nahranou smyčku v kamerovém záznamu, jde o činnost, která je časově náročná a přesahuje rozsah této práce. Je totiž nezbytné zpracovat větší množství kamerových záznamů, aby se předešlo *Overfittingu*, který byl popsán v první kapitole této práce. V případě hledání nahrané smyčky pouze z jednoho kamerového systému by tak mohlo dojít k situaci, že na této kameře bude algoritmus dolazen do funkční podoby, ale v případě použití na jiné kameře, v jiném prostředí, algoritmus nebude fungovat správně a celý proces se bude muset opakovat. Naopak v případě kratších kódů, jako je například *kontrola_pripojeni.py* bylo základem dodržet pravidla syntaxe kódu a doplnit do kódu správně IP adresu a port, na kterém vybraná kamera komunikuje.

ZÁVĚR

Tato práce začínala seznámením s nejčastěji používanými pojmy týkajících se umělé inteligence a vysvětlením jejich základních funkčních principů. Toto seznámení bylo nezbytné pro praktickou část této práce. V praktické části této práce došlo k představení programů určených pro video dohledové systémy. Seznámení s již existujícími programy bylo důležité k pochopení fungování těchto systémů a zjištění, ve kterých oblastech by mohly být navržené algoritmy přínosné. Poté následovalo navržení algoritmů samotných, majících za cíl zlepšit zabezpečení těchto systémů. Během návrhů došlo na základě analýzy chyb, selekce a dedukce k různým změnám v algoritmech, protože ne vždy se každý navržený postup setkal s úspěchem. Nicméně i tato část byla přínosnou neboť rozšířila autorovo povědomí o problematice. Poslední část práce, zaměřená na ověření funkcionality, ve své podstatě probíhá i v tomto momentě. Vzhledem ke komplexnosti kódů je potřeba věnovat navrženým algoritmům ještě další prostor pro jejich testovací provoz a optimalizaci. S přihlédnutím k tomu, že cílem práce byl návrh algoritmů, lze jejich testovací provoz brát jako „Proof of Concept“ a považovat tak cíle této práce za splněné.

SEZNAM POUŽITÉ LITERATURY

20 Years after Deep Blue: How AI Has Advanced Since Conquering Chess, 2023. *Scientific American: Science News, Expert Analysis, Health Research* [online]. Berlín: Springer Nature [cit. 2023-04-10]. Dostupné z: https://www.scientificamerican.com/article/20-years-after-deep-blue-how-ai-has-advanced-since-conquering-chess/

50 AI Terms Every Beginner Should Know: *TELUS International, 2023. TELUS International: Customer Experience & Digital Solutions* [online]. Vancouver: Telus International [cit. 2023-03-10]. Dostupné z: https://www.telusinternational.com/insights/ai-data/article/50-beginner-ai-terms-you-should-know

A Primer for understanding Reinforcement Learning, 2023. *TELUS International: Customer Experience & Digital Solutions* [online]. Vancouver: Telus International [cit. 2023-03-10]. Dostupné z: https://www.telusinternational.com/insights/ai-data/article/reinforcement-learning-primer

AGGARWAL, Charu C., [2018*]. Neural networks and deep learning: a textbook*. Cham: Springer. ISBN 3319944622.

Avigilon: Control Center 7 Soft [online], 2023. Vancouver: Avigilon Corporation [cit. 2023-02-28]. Dostupné z: https://www.avigilon.com/

Bezplatný online úvod do umělé inteligence pro každého: elementsofai.cz/ [online], 2023. Helsinki: MinnaLearn [cit. 2023-03-09]. Dostupné z: https://www.elementsofai.cz/

Bosch Video Management Software: Bosch Security and Safety Systems [online], 2023. Grasbrunn: Bosch Sicherheitssysteme [cit. 2023-02-28]. Dostupné z: https://www.boschsecurity.com/xc/en/solutions/management-software/bvms/

Dataversity: A Brief History of Deep Learning [online], 2023. Studio CIty: DATAVERSITY. Dostupné také z: https://www.dataversity.net/brief-history-deep-learning/#

Datový kurz PyLadies: Základy vizualizace - v pandas a pro pandas, 2023. *Nauč se Python!* [online]. [cit. 2023-03-09]. Dostupné z: https://naucse.python.cz/2020/pydata-praha-podzim/pydata/visualization_basics/

Deep learning pro segmentaci obrazu, 2023. *Technický týdeník* [online]. Praha: Business Media CZ [cit. 2023-04-10]. Dostupné z: https://www.technickytydenik.cz/rubriky/ict/deep-learning-pro-segmentaci-obrazu_42430.html

Deep Sentinel [online], 2023. Pleasanton: Deep Sentinel [cit. 2023-04-10]. Dostupné z: https://www.deepsentinel.com/

Genetec Inc.: Leader in unified physical security software [online], 2023. Montreal: Genetec [cit. 2023-02-28]. Dostupné z: https://www.genetec.com/

GitHub: Let´s build from here [online], 2023. San Francisco: Microsoft [cit. 2023-02-28]. Dostupné z: https://github.com/

GOODFELLOW, Ian, Yoshua BENGIO a Aaron COURVILLE, 2016. *Deep Learning* [online]. Cambridge: MIT Press [cit. 2022-11-01]. ISBN 0262035618. Dostupné z: https://www.deeplearningbook.org/

Home - OpenCV [online], 2023. Californie: OpenCV team [cit. 2023-03-09]. Dostupné z: https://opencv.org/

CHOLLET, François, [2018]. *Deep learning with Python.* Shelter Island, NY: Manning. ISBN 9781617294433.

Intrusion Detection System, 2023. *GeeksforGeeks* [online]. Noida: GeeksforGeeks [cit. 2023-04-09]. Dostupné z: https://www.geeksforgeeks.org/intrusion-detection-system-ids/

Intrusion Prevention System, 2023. *GeeksforGeeks* [online]. Noida: GeeksforGeeks [cit. 2023-04-09]. Dostupné z: https://www.geeksforgeeks.org/intrusion-prevention-system-ips/

ISpy: Open Source Camera Security Software [online], 2023. Margaret River: DeveloperInABox [cit. 2023-02-28]. Dostupné z: https://www.ispyconnect.com/

IVMS - 4200 Software - Hikvision: hikvision.com [online], 2023. Praha: Hikvision Czech [cit. 2023-02-28]. Dostupné z: https://www.hikvision.com/cz/products/software/ivms-4200/

JIRÁSEK, Petr, Luděk NOVÁK a Josef POŽÁR, 2015. *Výkladový slovník kybernetické bezpečnosti: Cyber security glossary* [online]. Třetí aktualizované vydání. Praha: Policejní akademie ČR v Praze [cit. 2023-03-07]. ISBN 978-80-7251-436-6.

KOLOUCH, Jan, 2016. *CyberCrime* [online]. Praha: CZ.NIC, z.s.p.o. [cit. 2023-03-07]. CZ.NIC. ISBN 978-80-88168-15-7.

KOLOUCH, Jan a Pavel BAŠTA, 2019. *CyberSecurity* [online]. Praha: CZ.NIC, z.s.p.o. [cit. 2023-03-07]. CZ.NIC. ISBN 978-80-88168-31-7.

Matematická biologie učebnice: Umělá inteligence: portal.matematickabiologie.cz [online], 2021. Brno: Masarykova Univerzita [cit. 2023-03-09]. Dostupné z:

https://portal.matematickabiologie.cz/index.php?pg=analyza-a-hodnoceni-biologickych-dat--umela-inteligence

MORAVČÍK, Matej et al., 2017. DeepStack: Expert-level artificial intelligence in heads-up no-limit poker. *Science* [online]. 356(6337), 508-513 [cit. 2023-03-11]. ISSN 0036-8075. Dostupné z: doi:10.1126/science.aam6960

MotionEyeOS [online], 2023. GitHub [cit. 2023-02-28]. Dostupné z: https://github.com/motioneye-project/motioneyeos/wiki

Nauč se Python! [online], 2023. Praha: online [cit. 2023-03-07]. Dostupné z: naucse.python.cz

Největší komerční dohledové centrum střední Evropy je v Praze, 2023. *Technický týdeník* [online]. Praha: Business Media CZ [cit. 2023-04-10]. Dostupné z: https://www.technickytydenik.cz/rubriky/denni-zpravodajstvi/nejvetsi-komercni-dohledove-centrum-stredni-evropy-je-v-praze_46627.html

Neuronové sítě - Elements of AI, 2023. *Bezplatný online úvod do umělé inteligence pro každého: elementsofai.cz* [online]. Helsinki: MinnaLearn [cit. 2023-03-09]. Dostupné z: https://course.elementsofai.com/cs/5

NĚMEČEK, Milan, 2008. *CCTV kamery a jejich využití v zabezpečení objektů*. Zlín.

Diplomová práce. Univerzita Tomáše Bati, Fakulta aplikované informatiky. Vedoucí práce

Milan Adámek

NIELSEN, Michael, 2019. *Neural Networks and Deep Learning. Neural Networks and Deep Learning* [online]. San Francisco: Michael Nielsen [cit. 2023-02-27]. Dostupné z: http://neuralnetworksanddeeplearning.com/

NumPy: naucse.python.cz, 2023. *Nauč se Python!* [online]. [cit. 2023-03-09]. Dostupné z: https://naucse.python.cz/lessons/intro/numpy/

PEDRYCZ, Witold a Shyi-Ming CHEN, ed., [2020]. *Deep learning: algorithms and applications. Cham: Springer. Studies in computational intelligence*. ISBN 978-3-030-31759-1.

Practical Python and OpenCV [online], 2020. 3. vydání. New York: PyImageSearch [cit. 2023-02-27]. Dostupné z:

https://minhtn1.github.io/Practical%20Python%20and%20OpenCV,%203rd%20Edition.pdf

PythonBooks - Python Wiki, 2023. *Python* [online]. Delaware: Python Software Foundation [cit. 2023-03-09]. Dostupné z: https://wiki.python.org/moin/PythonBooks

RUSSELL, Stuart J. a Peter NORVIG, 2010. *Artificial Intelligence: A Modern Approach [online]*. 3. vydání. Upper Saddle River: Prentice Hall [cit. 2023-03-09]. Prentice Hall series in artificial intelligence. ISBN 978-0-13-604259-4. Dostupné z: https://people.engr.tamu.edu/guni/csce421/files/AI_Russell_Norvig.pdf

Shinobi [online], 2023. Burnaby: Shinobi Systems [cit. 2023-02-28]. Dostupné z: https://shinobi.video/

Spyder IDE [online], 2023. Cambridge: Spyder Website Contributors [cit. 2023-03-18]. Dostupné z: https://www.spyder-ide.org/

SZELISKI, Richard, 2010. *Computer Vision: Algorithms and Applications*. London: Springer. Texts in computer science. ISBN 1848829345.

Software kamerových a dohledových systémů: Milestone [online], 2023. Chrudim: Bluecom s.r.o [cit. 2023-02-28]. Dostupné z: https://www.milestonesys.cz/

ŠALOMON, Zbyněk, 2021. *Komplexní zabezpečení objektu z hlediska ochrany a obrany.* Zlín. Bakalářská práce. Univerzita Tomáše Bati, Fakulta logistiky a krizového řízení. Vedoucí práce Jan Strohmandl.

TensorFlow [online], 2015. Mountain View: Google Brain [cit. 2023-03-10]. Dostupné z: https://www.tensorflow.org/

The Dawn of Robot Surveillance: AI, Video Analytics, and Privacy [online], 2019. New York: AMERICAN CIVIL LIBERTIES UNION [cit. 2022-10-31]. Dostupné z: https://www.aclu.org/report/dawn-robot-surveillance

Tkinter - Python interface to Tcl/Tk: Python 3.11.2 documentation, 2023. *Welcome to Python.org* [online]. Delaware: Python Software Foundation [cit. 2023-03-09]. Dostupné z: https://docs.python.org/3/library/tkinter.html

Tvorba grafického uživatelského rozhraní v Pythonu s využitím frameworku PySide: Root.cz. *Root.cz: Informace nejen ze světa Linuxu* [online]. Praha: Internet Info [cit. 2023-

03-07]. Dostupné z: https://www.root.cz/clanky/tvorba-grafickeho-uzivatelskeho-rozhrani-v-pythonu-s-vyuzitim-frameworku-pyside/

Úvod do neuronových sítí: automa.cz, 2005. *Automa: časopis pro automatizační techniku* [online]. Děčín: Automa, 2005(1) [cit. 2023-03-09]. Dostupné z: https://automa.cz/cz/casopis-clanky/uvod-do-neuronovych-siti-2005_01_30255_1330/

YOLO: Real-Time Object Detection [online], 2023. Seattle: University of Washington [cit. 2023-02-28]. Dostupné z: https://pjreddie.com/darknet/yolo/

What's the difference between CNN and RNN?, 2023. *TELUS International: Customer Experience & Digital Solutions* [online]. Vancouver: Telus International [cit. 2023-03-10]. Dostupné z: https://www.telusinternational.com/insights/ai-data/article/difference-between-cnn-and-rnn

Začátečnický kurz, 2023. *Nauč se Python!* [online]. [cit. 2023-03-09]. Dostupné z: https://naucse.python.cz/course/pyladies/

ZoneMinder [online], 2023. [cit. 2023-02-28]. Dostupné z: https://zoneminder.com/

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AI - umělá inteligence

API - aplikační programové rozhraní

CCTV - Closed Circuit Television

CNN - Konvoluční neuronové sítě

CPU - Central Processing Unit

DoS - Denial of service

DDoS - Distributed Denial of Service

GAN - Generativní adversární sítě

GPU - Graphical Processing Unit

GPT - Generative Pretrained Transformer

GUI - Grafické uživatelské rozhraní

IDE - Integrated Development Environment

IDS - Intrusion Detection System

IP - Internet Protocol

IPS - Intrusion Prevention System

LISP - List processing

MIT - Massachusettský technologický institut

MU - mimořádná událost

NLP - Natural language processing

PoE - Power over Ethernet

RBAC - Role-Based Access Control

RBF - Neuronové sítě s radiální bází

RFID - Radio Frequency Identification

RNN - Rekurentní neurónové sítě

ROI – Region of Interest

TCL - Tool Command Language

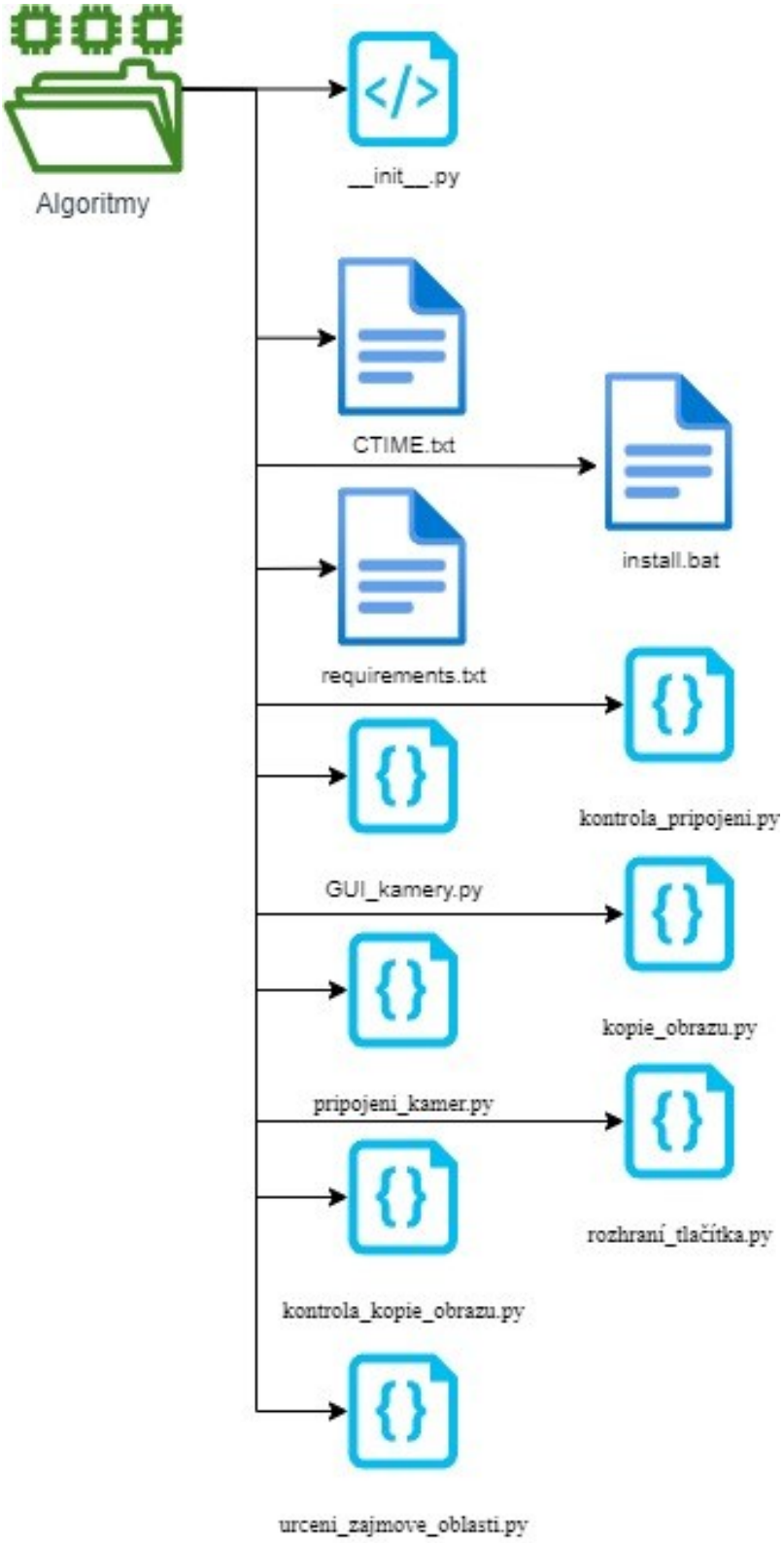
TPU - Tensor Processing Unit

VDS - video dohledový systém

SEZNAM OBRÁZKŮ

SEZNAM PŘÍLOH

PŘÍLOHA P I: INSTALAČNÍ SOUBOR S VYTVOŘENÝMI KÓDY



- V souboru CTIME.txt je napsán seznam kroků potřebných k správné funkci celého balíčku.
- __init__.py je iniciační soubor, který spouští vytvořené algoritmy.
- Install.bat se spustí v příkazovém řádku Windows a zařídí instalaci potřebných knihoven Pythonu.
- Requirements.txt je soubor ve kterém jsou sepsány požadované knihovny.
- Následující soubory s příponou .py jsou navržené algoritmy.